

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(Н И У « Б е л Г У »)**

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

**КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ**

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОПТИМИЗАЦИИ ПОСТАВОК
НА СКЛАД ООО "ЕВРОПА"**

Выпускная квалификационная работа
обучающейся по направлению подготовки 02.03.02 Фундаментальная
информатика и информационные технологии
очной формы обучения, группы 07001301
Звягинцевой Анастасии Сергеевны

Научный руководитель
к.т.н., доцент
Муромцев В.В.

БЕЛГОРОД 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ.....	4
1.1 Техничко-экономическая характеристика предприятия ООО «Европа».....	5
1.2 Техническое и программное обеспечение ТПК «ЕВРОПА».....	10
1.3 Анализ складского технологического процесса для разработки предложений по оптимизации склада.....	13
1.4 Процессы управления по оптимизации склада.....	14
1.5 Обоснование использования методов автоматизации для решения задачи оптимизации.....	18
ГЛАВА 2 КОНСТРУКТОРСКАЯ ЧАСТЬ.....	23
2.1 Транспортная задача и ее описание.....	23
2.2 Постановка задачи.....	23
2.3 Симплекс-алгоритм.....	25
2.4 Построение инфологической модели БД.....	30
2.5 Проектирование программного обеспечение задачи.....	36
2.6 Алгоритм написания программы.....	38
2.7 Описание процедур и программ.....	37
2.8 Руководство пользователей.....	40
ГЛАВА 3 ТЕСТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	43
ЗАКЛЮЧЕНИЕ.....	47
СПИСОК ЛИТЕРАТУРЫ.....	48
ПРИЛОЖЕНИЕ.....	50

ВВЕДЕНИЕ

Тема данной выпускной квалификационной работы «Автоматизированная система оптимизации поставок на склад».

Актуальность работы заключается в том, что в настоящее время уделяется усиленное внимание к проблеме максимального удовлетворения потребностей клиентов с приемлемыми для них затратами по сравнению с конкурентами. Сохранение качественных параметров продукции в значительной степени зависит от логистических процессов, которые существенно влияют также на себестоимость и цену товара.

Значение логистического процесса для конкурентной позиции предприятий считается неоспоримым, поскольку он - инструмент рационализации и снижения расходов предприятия. Однако стоит заметить, что глобальные расходы логистики составляют от 10 до 40% стоимости продаваемых товаров.

Функционирование логистической цепи начинается с составления заказа клиентом и заканчивается, когда покупатель заплатит за товар, который ему предоставили.

Логистическая цепь показывает движение продукта от поставщика через производителя и посредников к конечному потребителю. Одной из основных его частей считается состав, который есть практически на каждом его уровне.

Состав (в значении состава сырья и материалов для производства, и в значении склада для готовой продукции) считается интегральной частью цепи поставок, которую целесообразно рассматривать, с одной стороны, как место возникновения затрат, с другой - как элемент, который формирует доверие покупателя к продукту.

Если принимаем, что товаром считается то, что перемещается в логистической цепи, а транспорт считается тем, как товар перемещается, то

составом считается то, где он перемещается, т.е. это место, к которому и из которого перевозится товар. Следовательно, можно утверждать, что состав выполняет четыре главные функции: организацию приема, формирования, складирования, и организацию поставок. Эти функции позволяют содержать запасы на складе. Благодаря этому предприятие может обеспечить производство, а также фиксировать разницу между предложением и спросом как сырья и материалов, так и готовых продуктов.

При неудовлетворительной работе склада к основным рискам можно отнести повреждения товара, его потерю, долгое время загрузки и разгрузки, время подготовки поставки, большая трудоемкость, отсутствие места на складе, задержка производства или поставок по причине отсутствия соответствующей продукции. Поэтому оптимизация складских процессов позволяет обеспечить наличие необходимых для потребителей продуктов соответствующего качества по приемлемой цене.

Целью данной работы является разработка автоматизированной системы оптимизации логистических процессов и поставок на склад.

Объектом исследования данной выпускной квалификационной работы считается процесс поставки товаров на склад.

Предметом исследования считается оптимизация процесса поставки продукции на склады с помощью методов решения транспортной задачи.

В данной работе рассматриваются следующие задачи:

1. Изучение и исследование предметной области;
2. Разработка алгоритма оптимизации поставок;
3. Разработка моделей базы данных
4. Выбор инструментальных средств;
5. Реализация автоматизированной системы;
6. Тестирование разработанной системы.

Данная выпускная квалифицированная работа содержит 49 страниц, 18 рисунков, 6 таблиц и приложение с листингом программы.

ГЛАВА 1 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

1.1 Технико-экономическая характеристика предприятия ООО «Европа»

Торгово-производственная компания «Европа» ведет свою деятельность на рынке России с 2008 года.

Торговая сеть «Европа» включает в себя пятьдесят торговых предприятий и объектов в городах Черноземья, более 8000 рабочих мест и квалифицированных кадров, это более 500 000 покупателей в день.

Группа Компаний «Промресурс», ответственная за розничную торговлю через магазины торговой сети «Европа», занимается благотворительной деятельностью, регулярно выделяя денежные средства на ремонт больниц, школ и других социально-значимых объектов порядка 20 миллионов рублей в год.

Торговая сеть «Европа» находится в списке самых динамичных компаний центральной России. Организационно – правовая форма компании – индивидуальный предприниматель.

За время своего существования компания:

- Заключила долговременные договора с ведущими производителями, за счет постоянного роста ассортимента поставляемых товаров;
- Заработала авторитет на рынке реализации продукции промышленного и бытового назначения;
- Наладила сотрудничество с крупнейшими перевозчиками.

Расширение своего ассортимента - одно из главных направлений компании. Идя навстречу своим покупателям, она предлагает к продаже всё больший выбор востребованных на рынке товаров.

Основная задача оптово-розничной торговли заключается в снабжении товарами торговой розничной сети. Решение этой задачи связано с реализацией ряда торговых и технологических функций. К числу торговых функций, составляющих основу коммерческой деятельности оптовых предприятий, относятся:

- Изучение спроса населения и покупателей на товары;
- Заключение договоров поставки и контроль их исполнения;
- Взаимодействие с поставщиками с целью расширения ассортимента и увеличения выпуска высококачественных товаров, а так же производство новых;
- Формирование товарного ассортимента и управление запасами товаров;
- Заключение договоров купли-продажи с оптовыми покупателями и обеспечение контроля их исполнения;
- Оказание товарных услуг, связанных с оптовой продажей товаров;
- Организация и проведение рекламных мероприятий.

Технологические функции оптовых предприятий включают:

- Планомерный и бесперебойный завоз товаров на склады;
- Приёмку товаров по количеству и качеству;
- Хранение товаров в надлежащих условиях;
- Подготовка товаров к продаже (фасовку, упаковку, маркировку и т.п.);
- Обеспечение бесперебойного снабжения товарами розничной торговой сети.

Каждый сотрудник, участвующий в деятельности ТПК «Европа» на основе трудового договора, относится к тому или иному структурному подразделению [19].

При структуризации компании, ее деятельность рассматривается как совокупность бизнес-процессов. Бизнес-процесс подразделяется на четыре большие группы: основные, обеспечивающие, управление и развитие.

Организационная структура предприятия приведена на Рис.1.1.1.

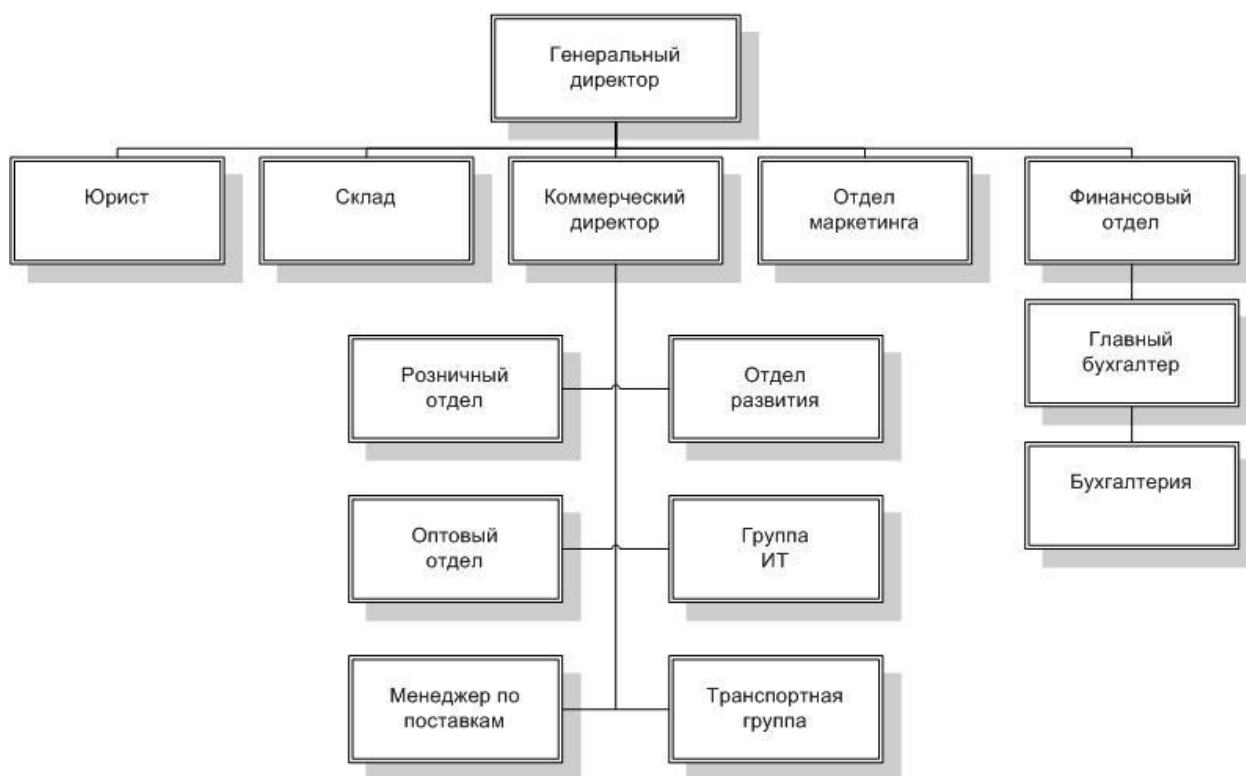


Рис.1.1. Организационная структура управления в компании «Европа»

Существующая организационная структура ТПК «Европа» имеет ряд проблем, как уже было сказано выше.

1. Решения представляются непоследовательными и произвольными, не соотнесенными со стандартами, правилами и критериями.
2. Сотрудники не понимают, чего от них хотят и каким образом определяется их вклад.
3. Отдельные работники становятся объектом конкурентного давления других сотрудников из-за отсутствия стандартных правил определения приоритетов.
4. Имеются сотрудники, слишком сильно нагруженные из-за того, что

они выполняют работы, которые следовало бы выполнять другим исполнителям.

5. Важная информация поступает с опозданием к сотрудникам, принимающим решения.

6. Наблюдается изолированность лиц, принимающих решения, в своих структурных подразделениях, что связано с отсутствием адекватных средств координации их деятельности.

7. Отмечается перегруженность сотрудников, принимающих решения, из-за их неспособности передать другим часть своих полномочий.

8. Отдельные сотрудники или группы сотрудников имеют конфликтные цели, не связанные с общей политикой организации и ее приоритетов.

9. В организации слишком много руководителей по сравнению с числом рядовых сотрудников.

10. Вышестоящие органы не поддерживают нововведения и изменения.

11. Отмечается недостаточная или неправильная координация деятельности сотрудников.

Проблемы структурного характера влекут за собой необходимость находить компромисс между справедливыми, но противоречащими друг другу требованиями:

- Потребность в надежности и контроле в ущерб гибкости и инициативе;
- Четкое разграничение полномочий и сфер ответственности в ущерб необходимости принимать во внимание информацию ряда заинтересованных лиц и штатных специалистов.

На данный момент уделяется усиленное внимание к проблеме максимального удовлетворения потребностей клиентов с приемлемыми для них затратами по сравнению с конкурентами. Сохранение качественных параметров продукции в значительной степени зависит от логистических

процессов, которые существенно влияют также на себестоимость и цену товара.

Необходимо воздействовать на факторы результативной работы членов трудового коллектива:

- ✓ разнообразие работы по содержанию;
- ✓ рост и расширение профессиональной квалификации работников;
- ✓ удобные графики работы;
- ✓ возможности проявления инициативы и осуществления самоконтроля;
- ✓ повышение ответственности сотрудников;
- ✓ продвижение по службе и т.д.

Компенсационный пакет предприятия включает:

- 1) базовые выплаты в виде оклада или тарифной ставки;
- 2) премия за качественное выполнение должностных обязанностей;
- 3) премия за достижение финансовых результатов;
- 4) доплаты за вредные условия труда;

Социальный пакет предприятия включает:

- 1) обязательное медицинское страхование;
- 2) обучения (повышение квалификации);
- 3) служебная машина;
- 4) мобильный телефон;
- 5) компенсация расходов, связанных с использованием личного автомобиля в служебных целях;

Действия по мотивации на предприятии ТПК «Европа» должны включать экономическое и моральное стимулирование для реализации потенциала сотрудников.

1.2 Техническое и программное обеспечение ТПК «ЕВРОПА»

В локальной вычислительной сети ТПК «ЕВРОПА» используется клиент-серверная архитектура, показанная на рис.1.2. [6,10].

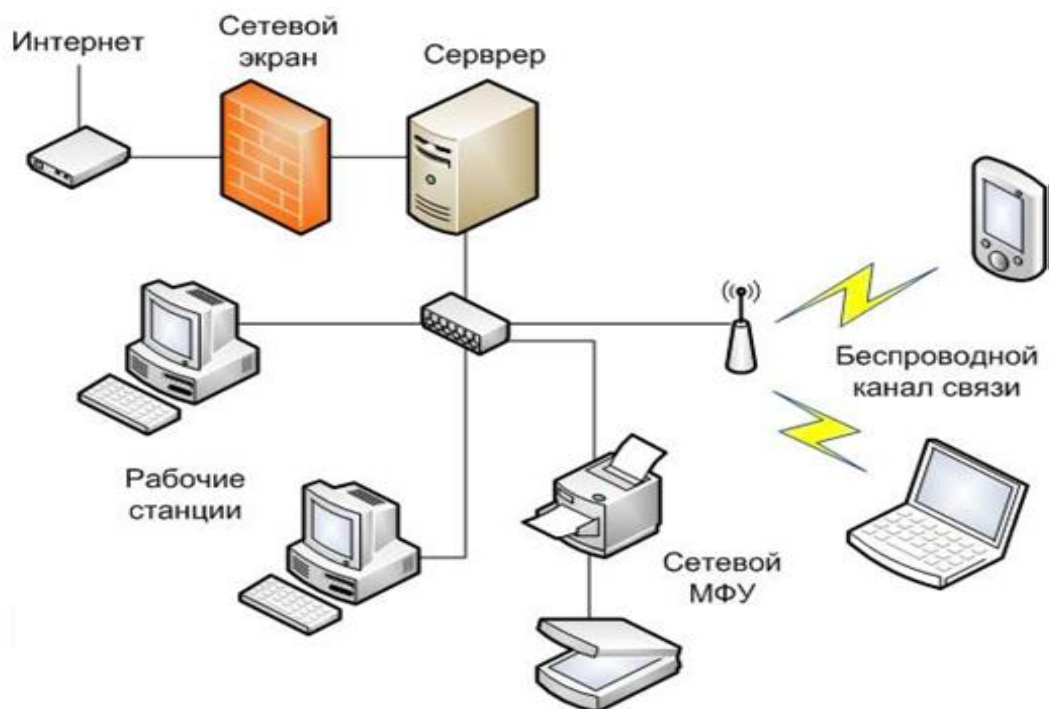


Рис.1.2. Техническая архитектура ТПК «ЕВРОПА»

В качестве активного оборудования ЛВС используется управляемый коммутатор 2-го уровня DES-1210-28 фирмы D-Link, представленный на рис.1.3.



Рис.1.3. Коммутатор D-Link , вид спереди

Коммутатор поддерживает полнодуплексный режим передачи данных и ориентирован на применение в топологии «звезда».

Серия коммутаторов D-Link DES-1210 включает в себя

модернизированные коммутаторы семейства Web Smart, объединяющие функции расширенного управления и безопасности, хорошую масштабируемость и производительность. Данные коммутаторы просты в использовании, оптимальны в плане соотношения цена/качество, являются недорогим и законченным решением для сетей малого и среднего бизнеса (SMB). Коммутатор DES-1210-28 управляется как с помощью протокола SNMP, так и на основе удобного Web-интерфейса

Модель DES-1210-28 поддерживает функцию Auto Voice VLAN, позволяющую устанавливать первоочередной приоритет для пакетов голосовой связи и обеспечивать бесперебойную работу приложения VoIP (передача голоса по сети), автоматическое определение MDI/MDIX, что исключает проблему применения кроссированных кабелей или портов UpLink и делает подключение коммутатора весьма необременительным.

Расширение стандартных функций 2-го уровня включает в себя IGMP Snooping, Spanning Tree (алгоритм отставного дерева), Port Mirroring и Link Aggregation Control Protocol (LACP), а управление потоком IEEE 802.3x позволяет напрямую подключить коммутатор к серверу для быстрой и надежной передачи данных.

Коммутатор поддерживает функцию D-Link Safeguard Engine, защищающую сеть от вирусного трафика, списки управления доступом (ACL) для повышения безопасности, аутентификацию на основе стандарта 802.1X, функцию предотвращения атак ARP Spoofing, а также DHCP Server Screening, запрещающую доступ неавторизованным DHCP-серверам.

Модель коммутатора выполнена в металлическом корпусе для установки в стандартную 19-дюймовую стойку и оснащена действенной пассивной системой охлаждения.

Для предоставления связи с интернетом в ЛВС организации используется ADSL-роутер известной фирмы TP-LINK модели TL-WR1043ND – устройство, сочетающее в себе модем, маршрутизатор (роутер) и беспроводную точку доступа (рис.1.4.).



Рис.1.4. Модем-роутер TP-LINK TL-WR1043ND

Соединенный напрямую с сервером и настроенный на автоматическое PPPoE-подключение, модем-роутер соединяется с телефонной линией через DSL-порт, при этом предоставляется постоянно активный высокоскоростной DSL-доступ в Интернет для всей сети. Данная модель имеет WAN-порт для подключения Ethernet-кабеля от Интернет-провайдера.

На персональных компьютерах пользователей установлена операционная система Microsoft Windows 7 (F2C-00545); 64-bit; Русский; 1 пользователь, DVD; BOX.

В своей работе сотрудники ТПК «ЕВРОПА» используют [4]:

- прикладное и офисное программное обеспечение: Программный пакет Microsoft Office 2013; Русский; 32-bit/64-bit; DVD; Retail (BOX);
- антивирусное программное обеспечение Kaspersky Lab Internet Security 2013 (KL1849LUBFS); Русский; 2 ПК; 1 год; диск B;
- в бухгалтерии установлена автоматизированная бухгалтерская система 1С: Предприятие 8.

Генеральный директор и главный бухгалтер контролируют обновление законодательных документов и статической информации полученных путем изучения Интернет источников и от органов государственной статистики.

Как было установлено, процессы ТПК «ЕВРОПА» автоматизированы. Для формирования заявок менеджер использует табличный процессор MS Excel 2013 [16].

1.3 Анализ складского технологического процесса для разработки предложений по оптимизации склада

Складской технологический процесс — это совокупность последовательно выполняемых операций, связанных с подготовкой к приему продукции по количеству и качеству, поступлением, перемещением, распаковкой, размещением на хранение, укладкой, отборкой, комплектацией, подготовкой к отправлению продукции потребителю.

Стратегии достижения конкурентных преимуществ предприятия на рынке достаточно полно освещены в научной литературе [5]. В частности, М. Портер [3] выделяет следующие стратегии достижения успеха в конкурентной борьбе: ценовое лидерство (по расходам), дифференциации и концентрации.

В [4] авторы рассматривают пять сил конкуренции и стратегии достижения преимуществ в зависимости от роли фирмы в конкурентной борьбе. В [2, с. 207] Е.В. Крикавский выделяет такие разновидности стратегий достижения конкурентных преимуществ, как лидерство по издержкам, развитие ключевых компетенций, кооперации и стратегии, ориентированные на время. К основным логистическим стратегиям автор относит стратегии минимизации логистических затрат, покупку (или продажу) логистических услуг, совершенствования логистической системы в масштабе многих фирм и быстрый цикл.

Сегодня, в условиях экономического кризиса, актуальной считается стратегия достижения лидерства по издержкам, которой соответствует стратегия снижения логистических затрат. В перспективе будет расти роль и стратегии дифференциации за счет улучшения уровня обслуживания клиентов.

Однако в научной литературе недостаточно внимания уделяется усилению конкурентных преимуществ предприятия за счет оптимизации

складских процессов в рамках реализации двух вышеупомянутых стратегий: лидерства по расходам и дифференциации.

Для разработки предложений по оптимизации складских процессов необходимо установить методы улучшения качественных показателей деятельности состава, определиться с выбором месторасположения распределительного центра в определенном регионе с целью оптимизации выполнения заказа потребителей и уменьшения количества торговых запасов.

1.4 Процессы управления по оптимизации склада

Управление складом настолько сложный процесс, что появилось много показателей, характеризующих деятельность состава.

Это показатели складских расходов, использование складской ротации товара, эффективности работников. Качественные показатели деятельности состава касаются убытков, которые возникли в результате плохого качества труда.

Внедрение системы управления качеством позволяет лучше понять факторы, связанные с качеством на складе. По этому подходу можно не только ограничить расходы состава, но и получить длительные конкурентные преимущества предприятия. Качественные факторы на складе можно разделить на три группы: факторы, связанные со временем, характеризующие изменчивость процесса, факторы, связанные с компетенцией работников, факторы, связанные с появлением убытков. Нахождение методов измерения и улучшение этих трех групп качественных факторов поможет управлять составом и в результате уменьшить расходы и получить конкурентное преимущество.

Большинство методов измерения факторов, связанных со временем, ограничивается определением средней величины, но она не характеризует изменчивость процесса, поэтому целесообразно применять методологию SPC

(Statistical Process Control), которая по своей сути предполагает изменчивость процесса.

Также под этим подразумевается проведение действий, предотвращающих возникновение потерь. На основании данных, полученных в текущем режиме, можно сделать выводы о предстоящем поведении процесса. Это преимущество SPC над другими инструментами обработки данных.

Применение SPC позволяет:

- Лучше узнать процессы и влияние различных факторов на их ход, что помогает отслеживать стабильность этих процессов;
- Результативнее диагностировать наличие ошибок еще до их проявления;
- Неоднократно улучшать системы управления без дорогостоящих инвестиций;
- Быстро принимать рациональные решения;
- Реально оценивать понесенные расходы;
- Управлять процессами в реальном времени;

Опыт показывает, что примерно 85% всех проблем основывается на системе управления. Остальные - это причины объективные (аварии, случаи и т.д.).

Эти 85% вызваны случайной переменной процесса системы, которая возникает из его природы, общего уровня технологии, организации, используемых методов деятельности и управления системой.

Следующим фактором, который влияет на качество складских процессов - а, следовательно, и на расходы – есть профессионализм работников.

Этот аспект очень важен, но часто его недооценивают. Одним из инструментов управления им считается "Система управления компетенциями работников" (SZKP). Этот инструмент позволяет планировать и оптимально подбирать людей на конкретные рабочие места. SZKP усиливает роль

руководителей самого низкого уровня в системе управления, делает текущий надзор за процессами, имеет решающее значение для результативности, эффективности и эластичности управления. Ведь чем результативнее является текущий надзор за процессом, тем меньше системных недочетов.

В результате внедрения SZKP растет эффективность использования потенциала работников. Дополнительным преимуществом этого метода считается то, что можно охватить всё предприятие. Риск можно исключить, контролировать, акцептовать или передать другой организации.

Процессный подход к риску позволяет рассматривать эту проблему в три этапа: угроза, непосредственно риск и потери. Угроза - фактор или ситуация, которая может привести к потере. Риск - это вероятность наступления события, влекущего возникновение ущерба. Потеря - повреждение товара, ошибочная поставка, производственный простой.

По такому делению можно создать карту рисков для принятия оптимального решения. Сущность метода заключается в сознательном принятии или исключении риска еще до возникновения негативных последствий.

Кроме понимания значения состава (узлового объекта) как прямого источника затрат, необходимо понимать его значение как прямого источника мультипликаций других логистических расходов (транспортных, информационных, риска и т.д.).

Поэтому одной из задач распределения, которую необходимо решить в целях улучшения уровня обслуживания клиентов, считается разработка моделей, связанных с выбором места расположения распределительного склада в регионе сбыта. Необходимость создания распределительного центра обуславливается, прежде всего, возможностью формирования "пакета заказов", причем поставлять товар после 100% заполнения автотранспорта. Таким образом, посчитается возможность сократить затраты поставки единицы товара.

Во-вторых, можно регулировать сроки заказов товара. Предприятие - производитель не требует создания больших запасов и ожидания отгрузки товара. Имея свой распределительный склад, производитель может организовать окончательный контроль продукции и поставлять покупателям только качественный товар.

И, наконец, в- третьих, можно организовать рецикл тары и сэкономить средства.

Различают два типа реализации складских проектов. Первый способ - компаниями, которые работают на рынке складской недвижимости, строят целый комплекс и ищут арендатора. Второй способ - это реализация складского проекта под заказчика, который может его брать в аренду и даже со временем выкупить, причем проект выполняется с учетом потребностей заказчика. Предыдущие переговоры между исполнителем и заказчиком очень важны, потому что не всегда клиент понимает, что именно он хочет получить.

Поэтому при разработке проекта состава необходимо учитывать долгосрочную стратегию заказчика. Для решения вопроса, где именно должен быть построен распределительный состав, необходимо учесть возможность покупки участка земли, которая не всегда доступна с точки зрения ее стоимости, существующей транспортной инфраструктуры, месторасположения и т.д.

Рынок складской недвижимости только развивается, и сегодня в России мало распределительных центров, которые бы предоставляли качественный сервис.

Требования клиентов к операторам на рынке логистических услуг, как показывает международный опыт, который касается: обработки большого количества наименований продукции, комплектации единицами, времени выполнения заказа (2-3 часа), количества повреждений (не более 0,01% в месяц), загрузки и разгрузки продукции (в пределах 0,5 часа), возможности адаптации информационной системы к системе заказчика. Такое качество

мало кто может обеспечить, поэтому сегодня многие производственные и посреднические предприятия занимаются непрофильным бизнесом - сами создают, оборудуют и обслуживают склады, имеют свой персонал и систему учета.

В западных странах уже почти отказались от непрофильного бизнеса, чтобы все свои усилия направить на основной вид деятельности - производство.

Главные проблемы, которые существуют у российских производителей и посредников, связанных со складированием, это: низкий уровень обслуживания; несовершенный учет; низкий контроль за товародвижением; устаревшая механизация и автоматизация; большой процент повреждений товара и ухудшения его качества; низкая скорость обработки грузов; отсутствие современной информационной системы, совместимой с системами всех субъектов логистической цепи [13].

1.5 Обоснование использования методов автоматизации для решения задачи оптимизации

Существует много внешних параметров и факторов, влияющих на выбор места локализации дистрибьюторского центра. Большинство авторов предметной литературы обращают особое внимание на величину перемещения товаров, места предоставления и приема грузов, а также длину трассы, которую преодолевает транспортное средство.

Кроме того, важным фактором, влияющим на локализацию такого центра, считается требование очень хорошего качества связи с сетью автомобильных и железных дорог.

Методы, свидетельствуют об отсутствии конкретных инструментов расчетов, необходимых при проектировании локализации дистрибьюторских центров.

Локализация такого центра должна быть подробно продумана, так как строительство такого объекта связано со значительными затратами, которые часто превосходят финансовые возможности.

Определенную исходную пользу составляет метод рангов при локализации дистрибьюторского центра.

Его целесообразно использовать для обоснования поэтапного стратегического развития дистрибьюторских сетей национальных производителей. Суть этого метода заключается в том, что систематизируются отдельные регионы (области) в отношении трех факторов, влияющих на логистику данного региона.

К этим факторам относятся: объем промышленного производства в регионе; численность населения; общее количество торговых заведений.

Метод установления рангов считается простым и быстрым методом, но его следует трактовать как один из вступительных при проектировании локализации логистических (дистрибьюторских) центров, где в равной степени уровень производства, как и население, а также количество магазинов являются важными параметрами, которые принимаются во внимание при локализации центра, но ими не следует ограничиваться.

К примененному методу после вступительной селекции следовало бы применить методы, которые учитывают величину перемещения грузов на трассах, расходы на транспортировку грузов, расстояния, которые преодолеваются транспортом, срок доступности товара и т.д.

Среди методов локализации складских объектов, учитывающих линейную функцию затрат, математическая запись имеет следующую форму:

$$ПВ = KB + kn + Mt, \quad (1.1)$$

где ПВ - полные затраты; kn - переменные удельные затраты на единицу продукции; Mt - величина продаж единиц продукции; KB - постоянные расходы.

Этот метод можно применить только для простых подсчетов. Над постоянными и переменными затратами можно полемизировать, так как не удастся точно оценить эти параметры в момент начальной локализации определенного складского объекта.

Поскольку предметная область, выбранная для дипломного проекта, затрагивает закупочно-продажную и учетную функцию данной компании, перечень входных и выходных документов будет достаточно велик. Автоматизация данных задач позволит ускорить документооборот, сократить ручные операции.

В таблице 1.1 представлены перечни и источники используемых входных документов.

Таблица 1.1

Перечни и источники используемых входных документов.

№ п/п	Перечень входных документов	Источники входных документов
1	Заказы	Покупатели
2	Приходные накладные	Поставщики

При условии, что клиент сделал заказ и хочет расплатиться наличными денежными средствами, менеджер проверяет наличие товаров, внесенных в список заказов, на розничном складе. Если товар имеется, то происходит процесс оформления расходной накладной или товарного чека.

Если же от клиента, который желает вести безналичный расчет, поступил заказ менеджеру, то он распечатывает счет на оплату и отдает его в бухгалтерию на хранение до момента оплаты. Затем оплаченный счет перенаправляется менеджеру с пометкой оплаты, после чего сотрудник сравнивает суммы заказа и оплаты.

Иногда клиент вносит сумму неравную сумме заказа. В таких случаях выясняется в бухгалтерии баланс клиента, и если он не совпадает с суммой заказа, то менеджер связывается с клиентом и выясняет причину. При этом

возможна корректировка заказа. Кроме того менеджер проверяет наличие товара, включенного в список заказов, на складе. А отдел поставок группирует полученные от клиентов заказы.

В отдел поставок приходные накладные по электронной почте присылаются поставщиками, откуда их пересылают в отдел продаж. После этого менеджер формирует поступление на оптовый склад товара путем его копирования из электронных накладных в таблицы Excel.

При условии возврата товара от клиента на склад, составляется приходная накладная. В таком случае происходит неверный расчет таких показателей результата, как выручка или прибыль, так как возврат товара приравнивается к его поступлению.

В Таблица 1. 2 1.2 представлены перечни и адресаты используемых результативных документов.

Таблица 1. 2

Перечни и адресаты результатных документов.

№ п/п	Перечень результатных документов	Адресаты результатных документов
1	Расходная накладная	Склад, бухгалтерия
2	Приходная накладная	Склад, бухгалтерия
2	Товарный чек	Покупатель
3	Счет на получение	Бухгалтерия

Расходная накладная оформляется менеджерами при отгрузке товара со склада. Распечатанный экземпляр отдается на склад для набора товара, после чего кладовщик приносит товар в торговый зал. Распечатанный экземпляр расходной накладной отдается в бухгалтерию только в том случае, если покупатель осуществил оплату безналичным расчетом. Он необходим для формирования бухгалтерских документов – накладных и счетов-фактур.

Счет на получение выдается покупателю при необеспеченном заказе для получения товара в оговоренный срок (если необходимого товара нет в настоящее время ни на одном из складов), только при покупке за наличный расчет. К нему прилагается кассовый чек, который подтверждает оплату заказа. Товарный чек выдается только по просьбе покупателя при отгрузке товара.

Некоторая информация, которая хранится в хранилищах, в дальнейшем подлежит обработке и выводу в информационной системе, в данном случае такой информацией являются входные показатели и результатные показатели, представленные в Таблица и Таблица 1. соответственно.

Таблица 1.3

Входные показатели

№ п\п	Наименование	Идентификатор
1	Закупочная цена	Цз
2	Процент скидки	С%
3	Процент наценки	Н%
4	Процент НДС	НДС%
5	Розничная цена	Цр
6	Оптовая цена	Цо

Таблица 1.4

Результатные показатели

№ п\п	Наименование	Идентификатор
1	Расчет розничной цены товара без скидки	$Ц_r = Ц_з * (1 + Н\% / 100) * (1 + НДС\% / 100)$
2	Расчет розничной цены товара со скидкой	$Ц_r = Ц_r - (С\% / 100) * Ц_r$
3	Расчет оптовой цены	$Ц_o = Ц_з - (С\% / 100) * Ц_з$

ГЛАВА 2 КОНСТРУКТОРСКАЯ ЧАСТЬ

2.1 Транспортная задача и ее описание

Основная задача транспортной логистики — перемещение требуемого количества товара в нужную точку оптимальным маршрутом за требуемое время и с наименьшими издержками. Различаются два варианта задачи: закрытая (или сбалансированная) и открытая модель (задачи с избытком или с дефицитом). В случае закрытой модели весь имеющийся в наличии груз развозится без остатка, что полностью удовлетворяет все потребности заказчиков (количество груза равно сумме заказов). В случае открытой модели либо все заказчики удовлетворены и при этом на некоторых складах могут остаться излишки груза (количество груза больше суммы заказов), либо весь груз оказывается израсходованным, хотя не все потребности полностью удовлетворены (количество груза меньше суммы заказов).

2.2 Постановка задачи

Известны c_{ij} , $i=1,2,...,m$; $j=1,2,...,n$ — стоимости перевозки единиц груза от каждого i -го поставщика каждому j -му потребителю. Переменными (неизвестными) транспортной задачи являются x_{ij} , $i=1,2,...,m$ $j=1,2,...,n$ — объемы перевозок от i -го поставщика каждому j -му потребителю.

Рассмотрим транспортную задачу:

Однородный груз сосредоточен у m поставщиков в объемах $a_1, a_2, ..., a_m$. Данный груз необходимо доставить n потребителям в объемах $b_1, b_2, ..., b_n$. Известны c_{ij} ($i=1,...,m$, $j=1,...,n$) — стоимости перевозки единицы груза от каждого i -го поставщика каждому j -му потребителю.

Требуется составить такой план перевозок, при котором запасы всех поставщиков вывозятся полностью, запросы всех потребителей удовлетворяются и суммарные затраты на перевозку всех грузов минимальны.

Исходные данные транспортной задачи представлены в таблице 2.1:

Таблица 2.1

Данные транспортной задачи

$\begin{matrix} & b_1 & b_2 & \dots & b_n \end{matrix}$	b_1	b_2	\dots	b_n
a_1	c_{11}	c_{12}	\dots	c_{1n}
a_2	c_{21}	c_{22}	\dots	c_{2n}
\dots	\dots	\dots	\dots	\dots
a_m	c_{m1}	c_{m2}	\dots	c_{mn}

Неизвестными транспортной задачи являются x_{ij} ($i=1,\dots,m, j=1,\dots,n$) - это объемы перевозок от каждого i -го поставщика каждому j -му потребителю. Они могут быть представлены в виде матрицы перевозок.

Составим математическую модель транспортной задачи в общем случае:

$$Z(x) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (2.1)$$

$$\sum_{j=1}^n x_{ij} = a_i, \quad i=1,\dots,m \quad (2.2)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j=1,\dots,n. \quad (2.3)$$

$$x_{ij} \geq 0, \quad i=1,\dots,m, j=1,\dots,n \quad (2.4)$$

Целевая функция задачи (2.1) определяет требование обеспечить минимум суммарных затрат на перевозку всех грузов.

Первая группа из m уравнений (2.2) описывает тот факт, что запасы всех m поставщиков вывозятся полностью. Вторая группа из n уравнений (2.3) выражает требование полностью удовлетворить запросы всех n потребителей.

Неравенства (2.4) являются условиями неотрицательности всех переменных задачи.

Таким образом, математическая формулировка транспортной задачи состоит в следующем: найти переменные задачи $X = (x_{ij})$, удовлетворяющие системе ограничений (2.2), (2.3), условиям неотрицательности (2.4) и обеспечивающие минимум целевой функции (2.1).

В рассмотренной модели транспортной задачи предполагается, что суммарные запасы поставщиков равны суммарным запросам потребителей, т.е.

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (2.5)$$

2.3 Симплекс-алгоритм

Геометрическая интерпретация, которая использовалась при решении задач линейного программирования (ЛП), перестает быть пригодной для этой цели при числе свободных переменных $n - m \geq 3$. Для нахождения решения задачи ЛП в общем случае (при произвольном числе свободных переменных) применяются не геометрические, а вычислительные методы. Из них наиболее универсальным является симплекс-метод.

Идея симплекс-метода относительно проста. Пусть в задаче ЛП имеется n переменных и m независимых линейных ограничений, заданных в форме уравнений, т.е. задача ЛП сформулирована в канонической форме.

Оптимальное решение, при условии его существования, достигается в одной из угловых точек, где $k = n - m$ из переменных равных нулю. Возьмем какие-либо k - переменных в качестве свободных и выразим через них остальные m - базисных переменных. Допустим, в качестве свободных выбраны первые $k = n - m$ переменных x_1, x_2, \dots, x_k , а остальные m выражены через них:

$$\begin{aligned} x_{k+1} &= a_{k+1,1}x_1 + a_{k+1,2}x_2 + \dots + a_{k+1,k}x_k + b_{k+1} \\ x_{k+2} &= a_{k+2,1}x_1 + a_{k+2,2}x_2 + \dots + a_{k+2,k}x_k + b_{k+2} \\ &\dots \\ x_n &= a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,k}x_k + b_n \end{aligned} \quad (2.6)$$

Предположим, что все свободные переменные x_1, x_2, \dots, x_k равны нулю, то решение будет называться базисным и иметь следующий вид:

$$x_1 = 0, x_2 = 0, \dots, x_k = 0, x_{k+1} = b_{k+1}, x_{k+2} = b_{k+2}, \dots, x_n = b_n.$$

Это решение может быть допустимым или недопустимым. Оно допустимо в том случае, если все свободные члены $b_{k+1}, b_{k+2}, \dots, b_n$ неотрицательны. Если это условие выполняется, то полученное решение называется допустимым базисным решением или опорным решением. Допустимое базисное решение соответствует одной из угловых точек ОДР (вершин ОДР). Предположим, что условие неотрицательности свободных членов выполнено. Тогда получится допустимое базисное решение, но является ли оно оптимальным? Чтобы проверить это, нужно выразить целевую функцию

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n, \quad (2.7)$$

которую требуется минимизировать, через свободные переменные x_1, x_2, \dots, x_k :

$$z = d_0 + d_1x_1 + \dots + d_kx_k, \quad (2.8)$$

Для этого надо в (2.2) подставить (2.1), чтобы выразить базисные переменные через свободные и привести подобные члены. Очевидно, что при $x_1 = 0, x_2 = 0, \dots, x_k = 0$ $z = d_0$, т.е. d_0 – это значение целевой функции при допустимом базисном решении. Следует проверить, возможно ли улучшить решение, т.е. уменьшить целевую функцию z , увеличивая какие-нибудь из переменных x_1, x_2, \dots, x_k (уменьшать их невозможно, т.к. все они равны нулю, а отрицательные значения переменных недопустимы). Если все коэффициенты d_1, d_2, \dots, d_k в формуле (2.3) положительны, то, увеличивая какие-то из переменных x_1, x_2, \dots, x_k сверх нуля, нельзя уменьшить целевую функцию z ; следовательно, найденное допустимое базисное решение является оптимальным.

Если же среди коэффициентов d_1, d_2, \dots, d_k в формуле (2.3) есть отрицательные, то, увеличивая некоторые из переменных x_1, x_2, \dots, x_k , а именно – те, коэффициенты при которых отрицательны, возможно улучшить решение, т.е. уменьшить z .

Пусть, например, коэффициент d_1 в формуле (2.7) отрицателен.

Значит, есть смысл увеличить x_1 , т.е. перейти от данного допустимого базисного решения (опорного решения) к другому, где переменная x_1 не равна нулю, а вместо нее равна нулю какая-то другая, т.е. перейти от одной вершины ОДР к другой вершине.

Увеличение x_1 “полезно” для целевой функции z , делает ее меньше.

Однако увеличивать x_1 надо аккуратно, так чтобы не стали отрицательными другие переменные $x_{k+1}, x_{k+2}, \dots, x_n$, выраженные через свободные переменные, в частности, через x_1 формулами (2.6).

Так как остальные свободные переменные x_2, \dots, x_k остаются равными нулю, то изменения базисных переменных $x_{k+1}, x_{k+2}, \dots, x_n$ от увеличения x_1 выражаются формулами:

$$\begin{aligned}
 x_{k+1} &= a_{k+1,1}x_1 + b_1 \\
 x_{k+2} &= a_{k+2,1}x_1 + b_2 \\
 &\dots \\
 x_n &= a_{n,1}x_1 + b_n
 \end{aligned}
 \tag{2.9}$$

Следует проверить, опасно ли для переменных $x_{k+1}, x_{k+2}, \dots, x_n$ увеличение x_1 , т.е. может ли оно делать их отрицательными? Да, опасно, если коэффициент при x_1 в соответствующем уравнении отрицателен. Если среди уравнений (2.8) нет уравнения с отрицательным коэффициентом при x_1 , то величину x_1 можно увеличивать беспредельно, а, значит, целевая функция z не ограничена снизу и оптимального решения задачи ЛП не существует.

Допустим, что это не так и что среди уравнений (2.9) есть такие, в которых коэффициент при x_1 отрицателен. Для переменных, стоящих в левых частях уравнений, увеличение x_1 опасно — оно может сделать их отрицательными. Необходимо взять одну из таких переменных x_1 и посмотреть, до какой степени можно все же увеличить x_1 , пока переменная x_1 не станет отрицательной?

1-ое уравнение из системы преобразуется в следующее (2.9).

$$x_1 = a_{l,1}x_1 + b_l$$

Здесь свободный член $b_l \geq 0$, а коэффициент $a_{l,1}$ отрицателен. Легко понять, что можно увеличивать x_1 только до значения, равного $-b_l / a_{l,1}$, а при дальнейшем увеличении x_1 переменная x_1 станет отрицательной.

Берется из переменных $x_{k+1}, x_{k+2}, \dots, x_n$ та, которая раньше всех обратится в нуль при увеличении x_1 , т.е. та, для которой величина $-b_l / a_{l,1}$ меньше всего. Пусть этой переменной будет x_r . Тогда имеет смысл “перерешить” систему уравнений (2.7) относительно базисных

переменных, выведя из числа свободных переменных x_1 и переведя вместо нее в группу свободных переменных x_r .

Действительно, нужно перейти от допустимого базисного решения, задаваемого равенствами $x_1 = 0, x_2 = 0, \dots, x_k = 0$, к допустимому базисному решению, в котором уже $x_1 \neq 0$, а $x_2 = 0, \dots, x_k = 0, x_r = 0$.

Первое допустимое базисное решение получится, если обратить в нуль все новые свободные переменные x_1, x_2, \dots, x_k ; второе - если обратить в нуль прежние переменные x_2, \dots, x_k, x_r . Базисными переменными при этом будут $x_1, x_{k+1}, \dots, x_{r-1}, x_{r+1}, \dots, x_n$.

Предположим, уравнения типа (2.1) для нового набора базисных и свободных переменных составлены. Тогда можно выразить через новые свободные переменные и целевую функцию z .

Если все коэффициенты при переменных в этой формуле положительны, то в итоге получилось относительное решение (в нем все свободные переменные положены равными нулю).

Если среди коэффициентов при переменных есть отрицательные, то процедура улучшения решения продолжается: система вновь перерешается относительно других базисных переменных, и так далее, пока не будет найдено оптимальное решение, обращающее целевую функцию z в минимум, или не будет установлено, что целевая функция не ограничена снизу, т.е. задача ЛП не имеет решения.

Изложенное выше является содержанием симплекс-алгоритма. Суть симплекс-алгоритма состоит в установлении является ли допустимое базисное решение оптимальным. Если нет, то в указании как перейти к новому допустимому базисному решению, в котором значение целевой функции будет не больше, чем в предыдущем решении.

2.4 Построение инфологической модели БД

Построение инфологической модели БД следует начать с реализации диаграммы потоков данных (DFD). Стандарт описания бизнес-процессов DFD — Data Flow Diagram переводится как диаграмма потоков данных и используется для описания процессов верхнего уровня и для реально существующих в организации потоков данных. Созданные модели потоков данных компании могут быть использованы при решении таких задач, как:

1. определение существующих хранилищ данных (текстовые документы, файлы, система управления базой данных — СУБД);
2. определение и анализ данных, необходимых для выполнения каждой функции процесса;
3. подготовка к созданию модели структуры данных организации, так называемая ERD-модель (IDEF1X);
4. выделение основных и вспомогательных бизнес-процессов организации.

Диаграммы потоков данных показывают, как каждый процесс преобразует свои входные данные в выходные, и выявляют отношения между этими процессами. DFD представляет моделируемую систему как сеть связанных работ.

Существуют две нотации диаграмм: графические нотации Йордана Де Марко и Гейна-Сарсона. Различие заключается в графических формах объектов. [2]

Главные объекты нотации рис.2.1.:

- Процессы (Activities). Отображают процессы обработки и изменения информации;
- Стрелки (Arrows). Отображают информационные потоки;
- Хранилища данных (Data Store). Отображают данные, к которым осуществляется доступ, эти данные используются, создаются или изменяются;

– Внешние сущности (External References). Отображают объекты, с которыми происходит взаимодействие.

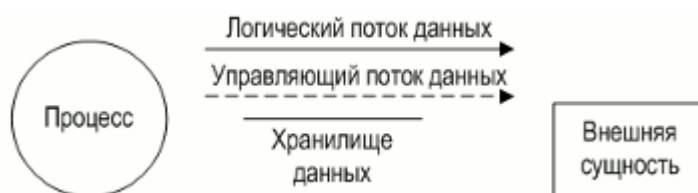


Рис.2.1. Главные объекты DFD в нотации Йордана Де Марко

На **Error! Reference source not found.**показана диаграмма потоков данных в рамках рассматриваемой предметной области компании.

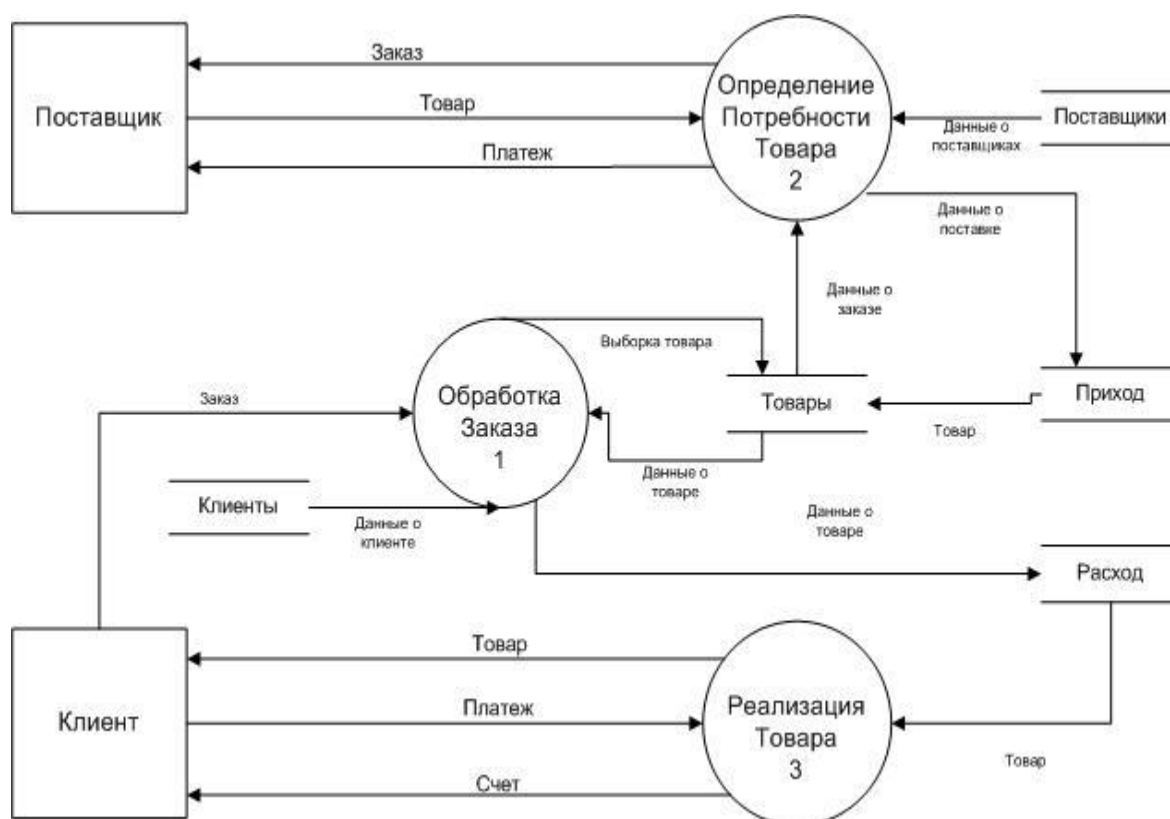


Рис.2.2. Диаграмма потоков данных

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представление той информации, которую предполагается хранить в создаваемой базе данных.

Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком, однако он не может быть использован в

чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты) [3].

Разгрузка товара. Это первый этап поступления товара на склад. Для того чтобы реализовать поставку на склад, необходимо знать кому нужно сделать заказ на нужный товар, следовательно, можно определить сущность «Поставщики», хранящую в себе в качестве атрибутов информацию о предприятии (наименование предприятия, адрес, телефон, электронная почта, а также различные банковские и юридические реквизиты). Так как поставщики поставляют нам товар, то определяется сущность «Приход», атрибуты сущности хранят информацию о поставке (наименование товара, дата поставки, количество товара и цена товара).

На рис.2.3. показана инфологическая модель, включающая в себя сущности и атрибуты сущностей. Инфологическая модель базы данных представляет собой описание объектов (сущностей), с набором атрибутов и связей между ними, которые выявляются в процессе исследования как входных, так и выходных данных.

Она предназначена для структурного образования предметной области, с ориентированием на информационное внимание пользователей, разрабатываемой системы. Так же инфологическая модель должна быть как стабильной, так и неизменной, и являться представлением аспекта пользователя на описанную предметную область.

Однако, при проектировании инфологической модели, должна присутствовать возможность для её увеличения и вставки вспомогательных данных [1]

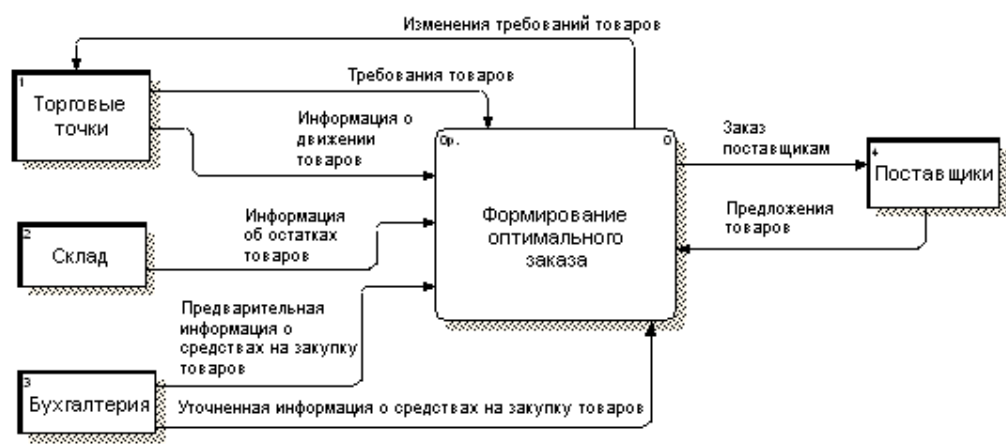


Рис.2.3. Инфологическая модель данных

Сегодня наиболее распространены реляционные модели данных. Любая система данных какой-либо сложности может быть сведена к набору таблиц (или "отношений" в терминологии СУБД). Каждое отношение (таблица) может быть представлено в виде прямоугольного массива со следующими свойствами:

1. Каждая ячейка в таблице представляет точно один элемент данных;
2. Каждая таблица имеет однородные столбцы, все элементы в любом из столбцов одного и того же вида;
3. Каждому столбцу назначено определенное имя;
4. Все строки различны, дублировать строки не разрешается;
5. Строки, и столбцы не зависят от последовательности, просмотр в различной последовательности не может изменить информационное содержание отношения;
6. Каждая строка олицетворяет уникальный элемент данных, который ею и описывается;
7. Столбцы представляют собой отдельные куски информации (атрибуты данных), которые известны о данном элементе.

Вообще, лишь немногие реальные базы данных могут быть описаны при помощи единственной таблицы. Большинство приложений используют множество таблиц, которые содержат столбцы (поля) с одинаковым именем. Эти общие данные позволяют, объединяя две (или несколько) таблицы, строить осмысленные ассоциации.

Нормализация — это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости.

Существует несколько правил нормализации баз данных. Каждое правило называется «нормальной формой».

Сущность первой нормальной формы:

- Устранение повторяющихся групп в отдельных таблицах;
- Создание отдельной таблицы для каждого набора связанных данных.
- Идентификация каждого набора связанных данных с помощью первичного ключа.

Если выполняется первое правило, говорят, что база данных представлена в «первой нормальной форме».

Сущность второй нормальной формы:

- Создание отдельных таблиц для наборов значений, относящихся к нескольким записям.
- Связка данной таблицы с помощью внешнего ключа.

Если выполняется второе правило, говорят, что база данных представлена во «второй нормальной форме».

- Третья нормальная форма
- Устранение поля, не зависящие от ключа.

Если выполняются три первых правила, считается, что база данных представлена в «третьей нормальной форме».

Разрабатываемая БД нормализована до 3НФ и представлена на **Error!**
Reference source not found. рис.2.4.

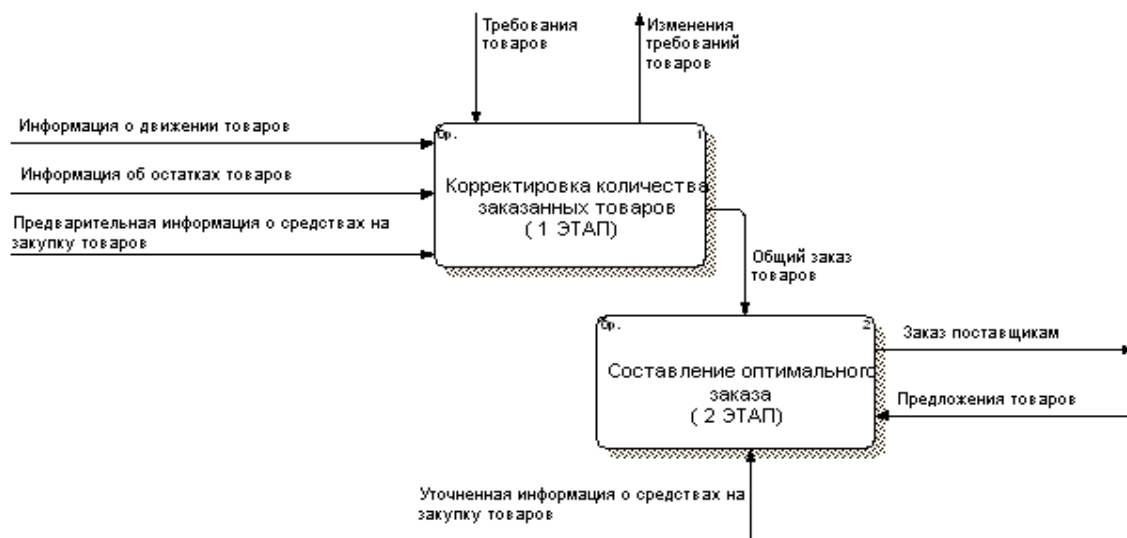


Рис.2.4. Этапы формирования оптимального заказа

К подготовительным мероприятиям по приему товаров на складе относится:

- установление места разгрузки транспортных средств, максимально приближенного к помещению для хранения;
- определение необходимого количества работников для разгрузки и распределение работ между ними;
- определение необходимого количества и видов подъемно-транспортного оборудования и подготовка его к прибытию товаров;
- заблаговременное определение мест хранения;
- подготовка документации, связанной с оформлением приемки-сдачи товаров.

Прежде чем приступить к непосредственному приему товара, необходимо его разгрузить. При разгрузке осуществляется передвижение товаров с автотранспорта на товарную платформу.

Разгрузку осуществляют работники того предприятия, чьим автотранспортом был доставлен товар на склад, то есть это могут быть работники фирмы поставщика, транспортной организации, фирмы покупателя. Если товар доставлен собственным автотранспортом фирмы покупателя или железной дорогой, разгрузку осуществляют работники склада.

2.5 Проектирование программного обеспечения задачи

Схемы "Дерево функций" и "Сценарий диалога" по своей сути непосредственно позволяют наглядно показать иерархию функций управления оптимизацией и обработки данных, которые автоматизированы в разработанном программном продукте.

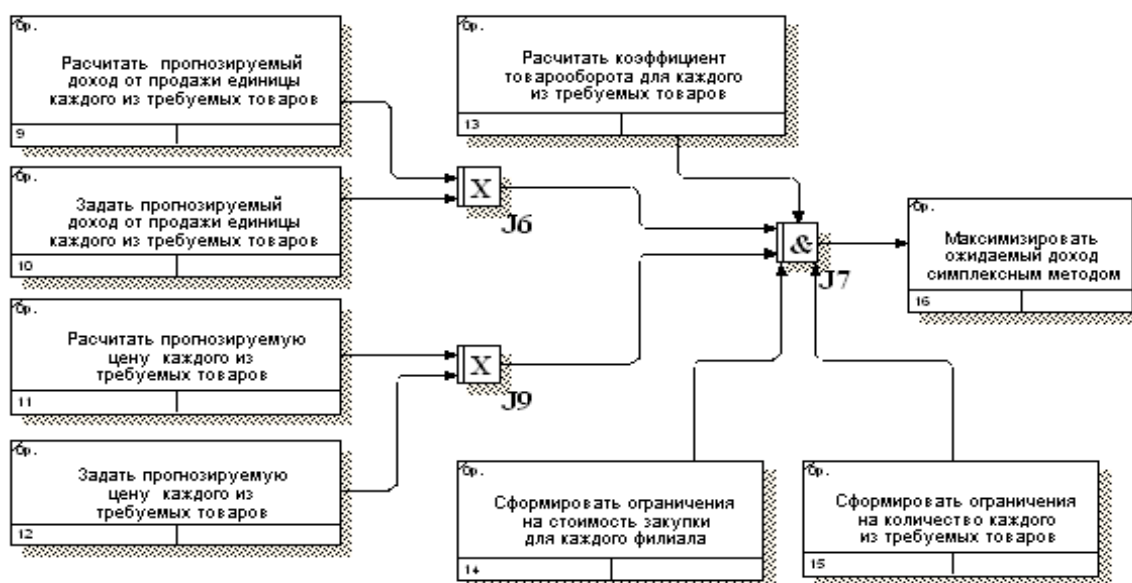


Рис.2.5. Сценарий диалога

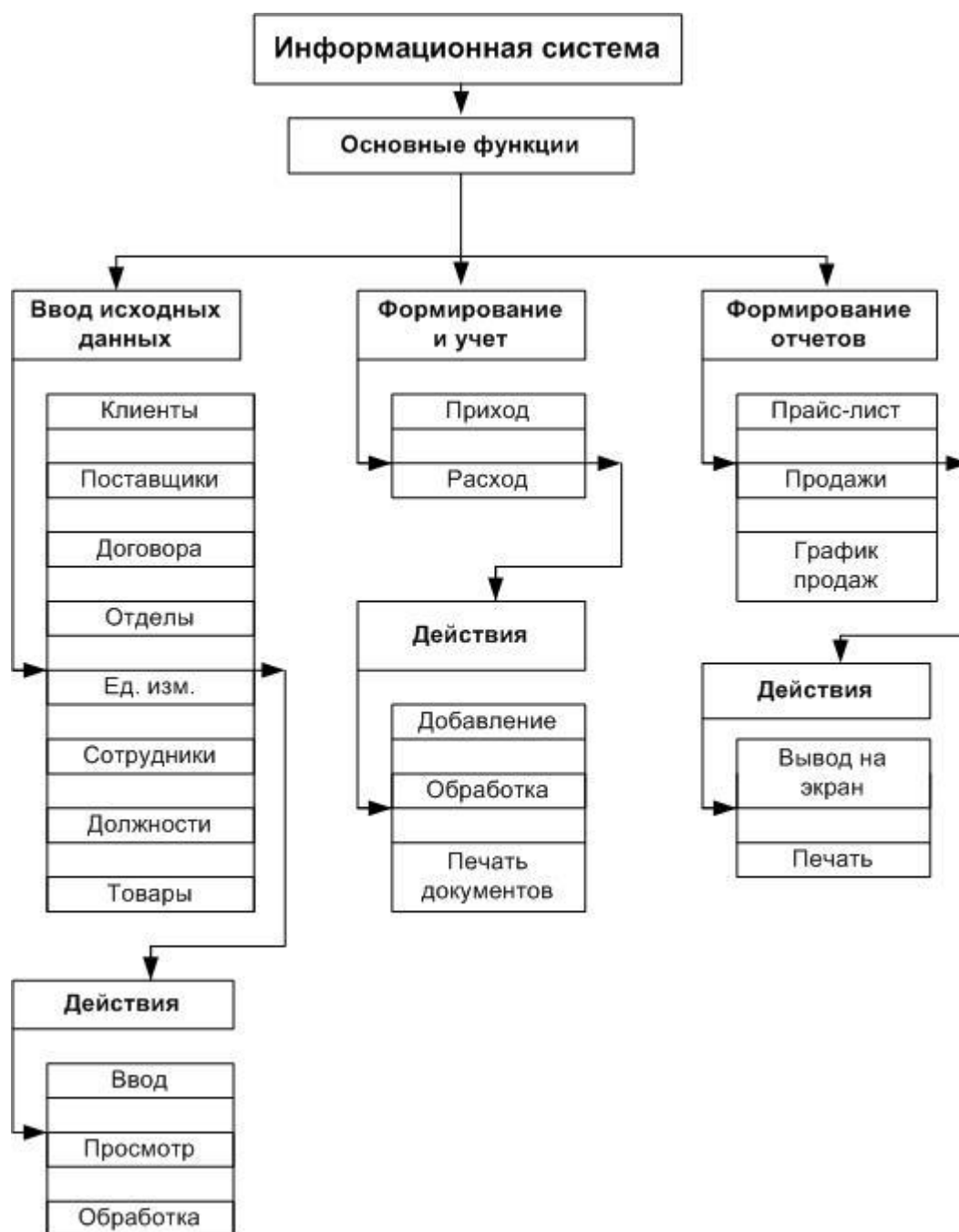


Рис.2.6. Дерево функций

2.6 Описание процедур и программ

Краткое описание программных модулей представлено в таблице 2.1

Структурную схему пакета удобно непосредственно представить в виде рис.2.6.

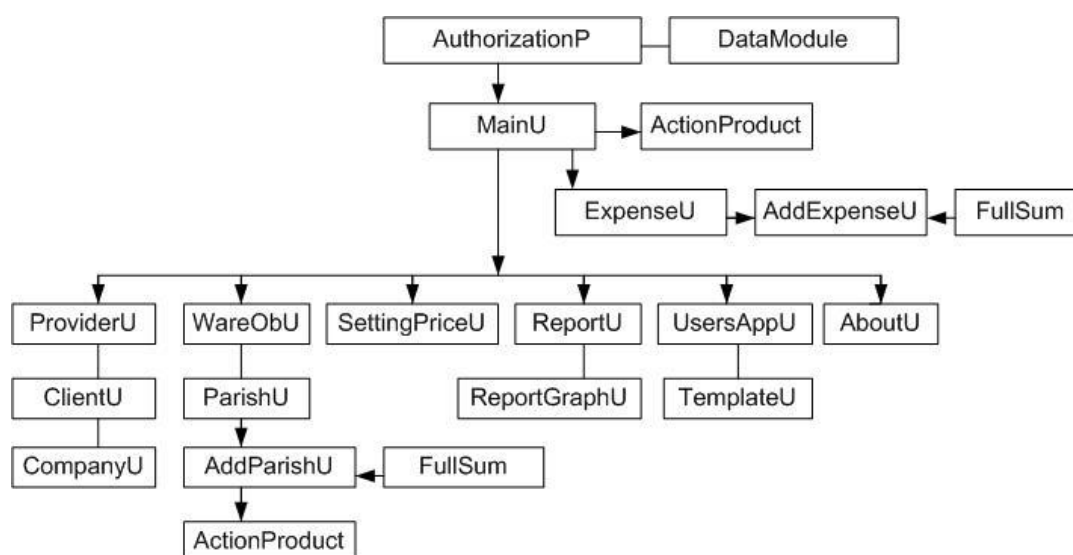


Рис.2.6. Структурная схема пакета

Таблица 2.1

Описание программных модулей

AuthorizationP1	Окно входа в систему ПО
ActionProduct1	Окно добавления, изменения товара ПО
AboutU1	Окно информации «О программе» ПО
ExpenseU1	Окно просмотра расходов ПО
AddExpenseU1	Окно добавления расхода ПО
ParishU1	Окно просмотра приходов ПО
AddParishU1	Окно добавления прихода ПО
ClientU1	Окно для работы с покупателями ПО
CompanyU1	Окно для работы с данными о компании ПО
DataModule1	Контейнер для хранения элементов ПО соединения проекта программного приложения с хранилищем данных
FullSum1	Модуль для перевода денежных значений в пропись ПО
MainU1	Главное окно приложения ПО
ProviderU1	Окно для работы с поставщиками ПО
ReportGraphU1	Окно вывода графика продаж ПО

Продолжение. Таблица 2.1

Описание программных модулей

ReportU1	Окно отчета о продажах ПО
SettingPriceU1	Окно изменения наценки, НДС ПО
TemplateU1	Окно для работы с шаблонами ПО
UsersAppU1	Окно для работы с пользователями и должностями ПО
WareObU1	Окно для работы с отделами, группами и ед. изм. ПО

2.7 Алгоритм работы программы

После запуска программы вводятся данные, затем происходит проверка задачи на закрытость/открытость. После чего начинается поиск оптимального решения задачи, если задача не оптимальна, производится поиск ключевого элемента, ключевой строки и ключевого столбца в матрице, затем - нахождение новых элементов задачи и построение таблицы. Далее возвращаемся к поиску оптимального решения, если задача вновь не оптимальна, то повторяются проделанные ранее шаги, если оптимальна, то происходит останов. Ниже представлен алгоритм в виде блок-схемы (см. рис.2.7.).



Рис.2.7. Алгоритм работы программного комплекса

2.8 Руководство пользователей

Руководство пользователя предназначено для того, чтобы научить пользователя программы работать с ней. При этом предполагается, что пользователь имеет некоторый начальный навык работы с компьютером, т.е. умеет работать с клавиатурой и мышью, ориентируется в сообщениях выдаваемых компьютером, знает, что такое Windows, окно и т.п.

Информационная система разработана с использованием программного средства визуальной разработки – C++.

Схема работы программы представлена на рис.2.8.:

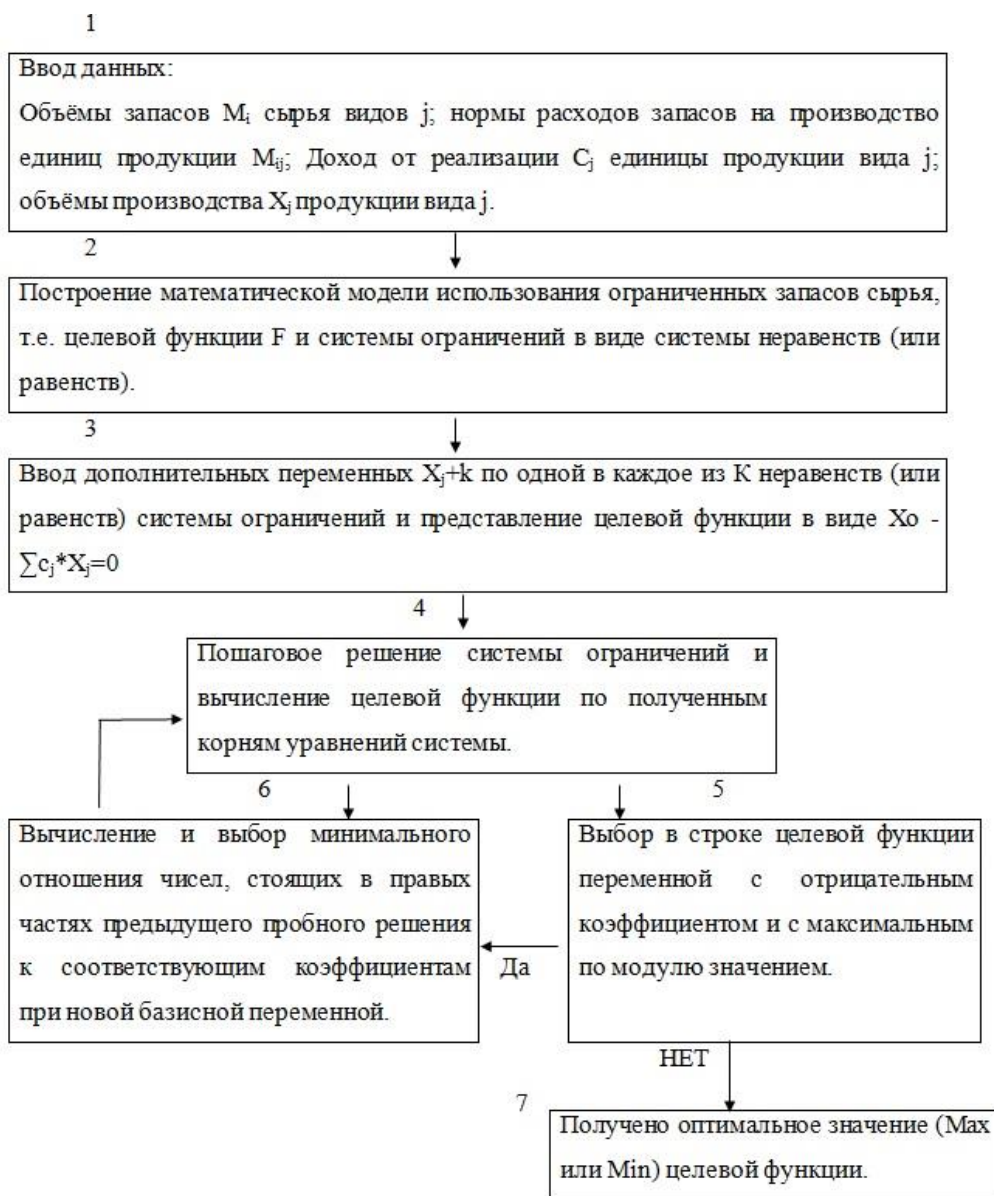


Рис.2.8. Схема работы программного комплекса

База данных FireBird разработана при помощи программы IVExpert.

Информационная система рассчитана на работу под управлением ОС XP, Vista, Seven. Для работы программы, необходимо чтобы на компьютере были установлены сервер Firebird версии 2.5, IVExpert, а также клиентское приложение (разработанная ИС).

Необходимость создания распределительного центра обуславливается прежде всего возможностью формирования "пакета заказов", причем поставлять товар после 100% заполнения автотранспорта.

Таким образом, посчитается возможность сократить затраты поставки единицы товара.

При установке системы необходимо зарегистрировать пользователей системы, назначить права и установить пароль.

Регистрация пользователей, назначение прав и ролей происходит в IVExpert, после этого нужно прописать путь к базе данных в файле конфигурации (это должен делать администратор системы), далее необходимо данного пользователя зарегистрировать в ИС, добавив должность сотрудника в компании.

Поэтому одной из задач распределения, которую необходимо решить в целях улучшения уровня обслуживания клиентов, считается разработка моделей, связанных с выбором места расположение распределительного склада в регионе сбыта.

После этого заполнить справочники в системе, добавить пути к шаблонам документов и отчетов, установить необходимые цены, и можно приступать к работе.

Вверху окна размещено меню. Внизу информационная панель с датой, временем и именем пользователя работающего с программой.

Ниже, на рис.2.9, приведена логическая модель базы данных.

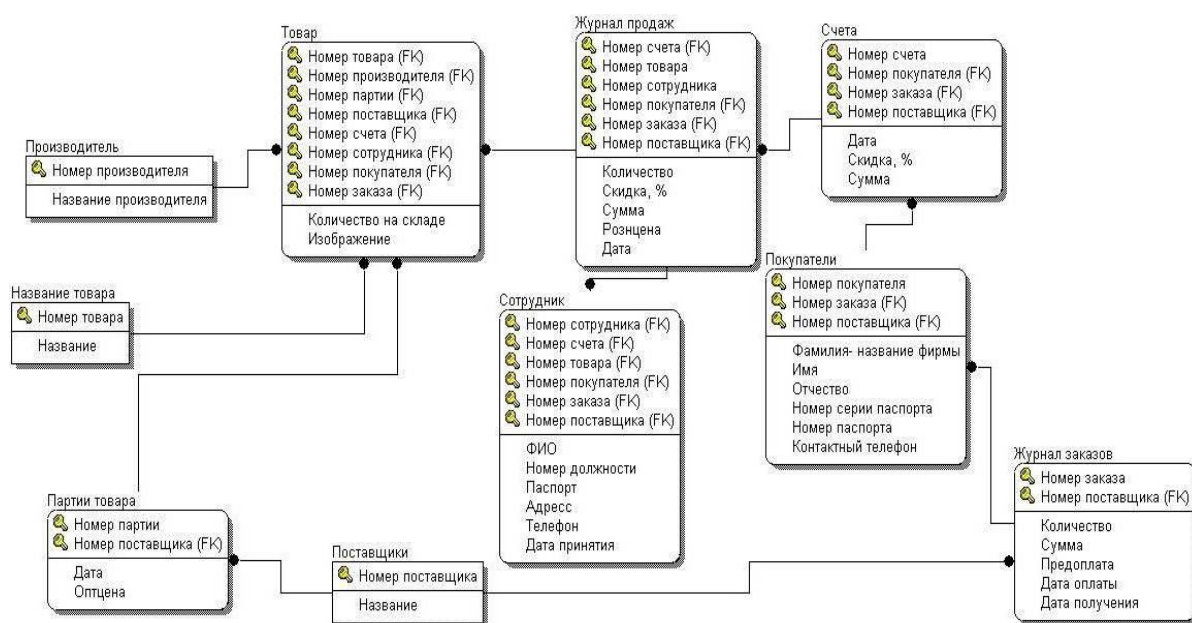


Рис.2.9. Логическая модель БД

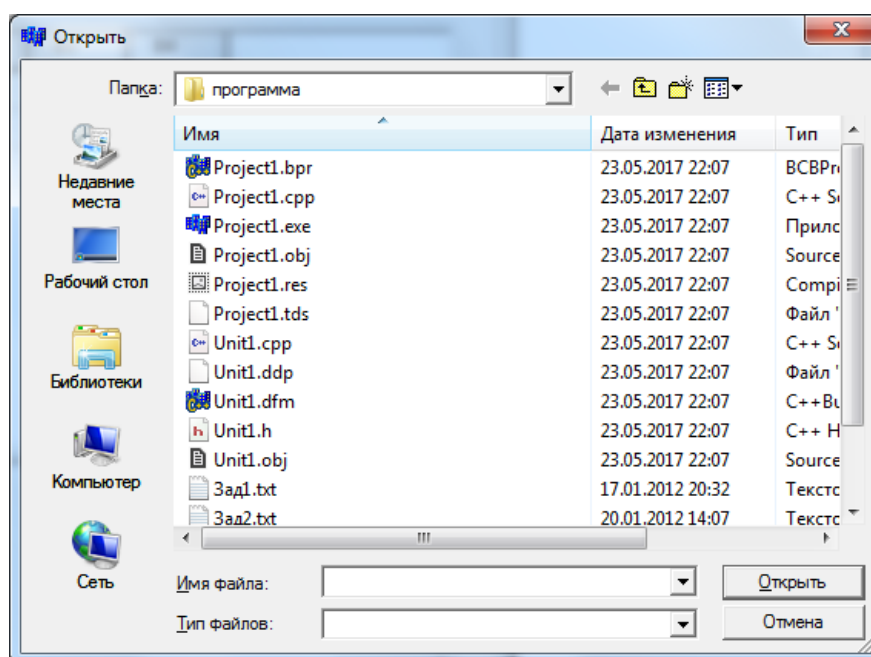


Рис.3.2. Окно добавления данных

После заполнения полей данными и нажатия на кнопку «Рассчитать», определяется состояние транспортной задачи, после чего появляется окно, оповещающее об этом состоянии, показанное на рис.3.3. и рис.3.4.

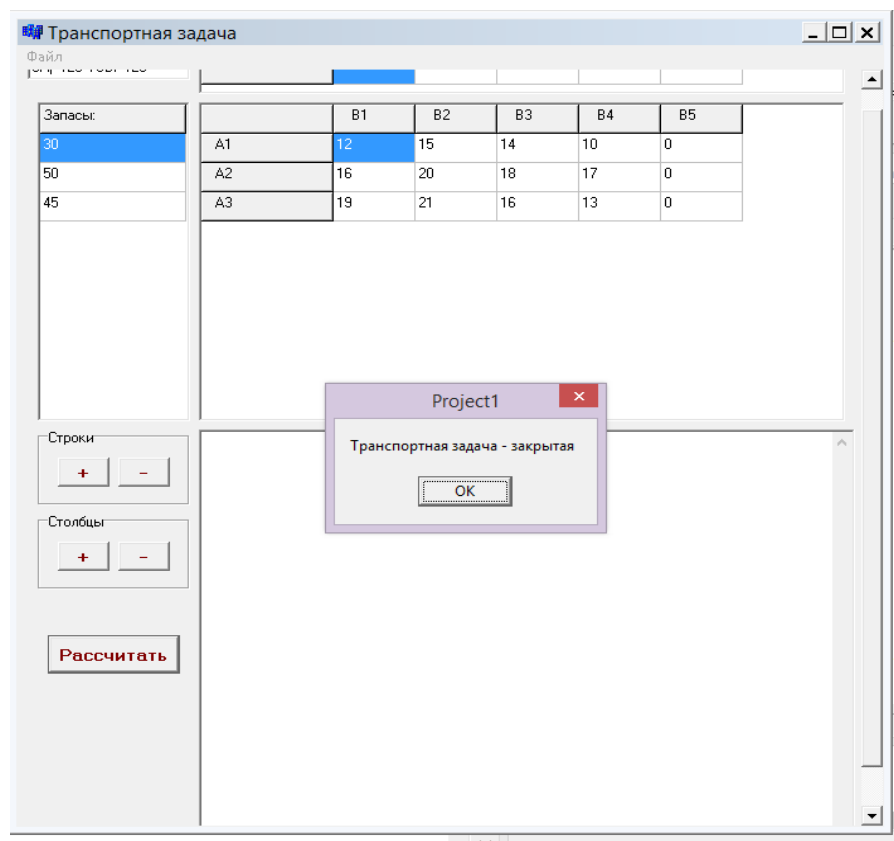


Рис.3.3. Состояние закрытой транспортной задачи

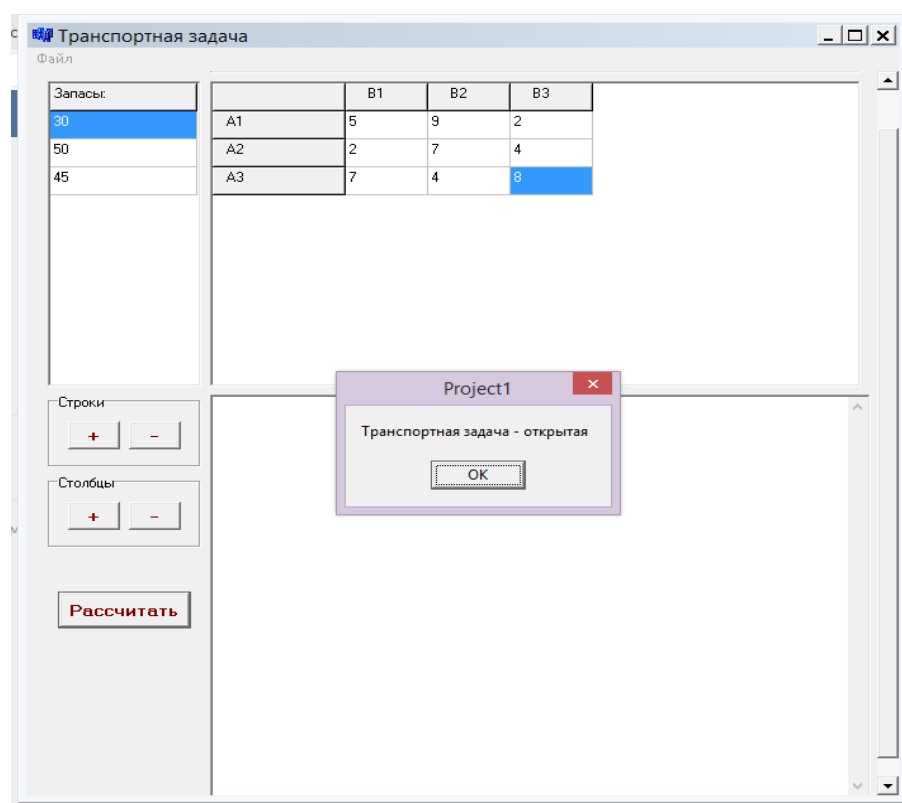


Рис.3.4. Состояние открытой транспортной задачи

В случае, если транспортная задача оказалась открытой, программа преобразует ее в закрытую путем добавления фиктивного потребителя. Данный случай представлен на рис.3.5.

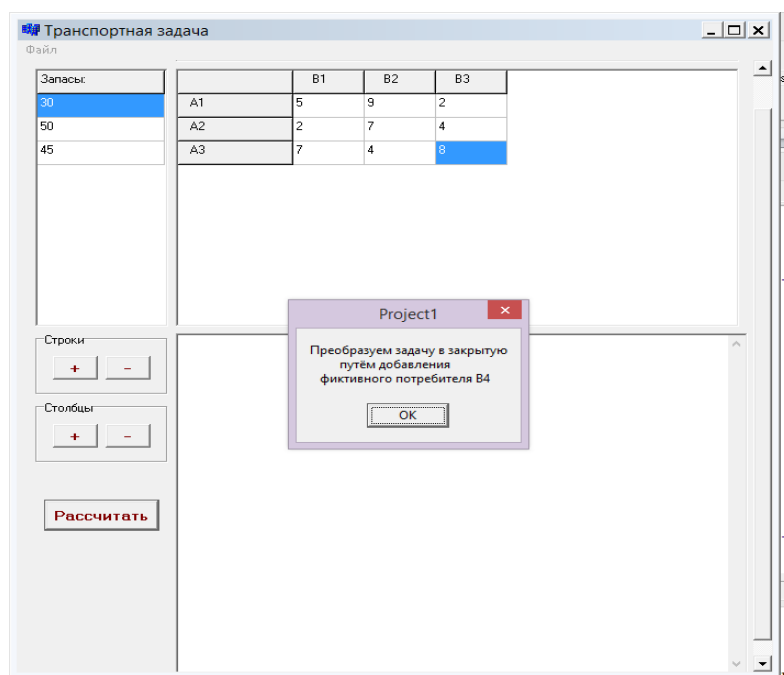


Рис.3.5. Преобразование транспортной задачи

Затем программа производит расчёты для нахождения оптимального плана, т.е. для нахождения минимальных транспортных затрат. Пример данных расчетов представлен на рис. 3.6.

Транспортная задача

Файл

Запасы:		B1	B2	B3	B4	B5
30	A1	12	15	14	10	0
50	A2	16	20	18	17	0
45	A3	19	21	16	13	0

Строки: + -

Столбцы: + -

Рассчитать

F=1810

Опорный план после устранения вырожденности:

```
-1 -1 -1 30 -1
20 25 0 -1 5
-1 -1 35 10 -1
```

Потенциалы:

u: 0 5 3

v: 11 15 13 10 -5

Матрица оценок свободных ячеек (если ячейка занята - ставим 88)

```
1 0 1 88 5
88 88 88 2 88
5 3 88 88 2
```

Оптимальный план:

```
-1 -1 -1 30 -1
20 25 0 -1 5
-1 -1 35 10 -1
```

Fmin=1810

Рис.3.6. Окно расчета

При решении данной задачи было необходимо использовать теорию размытых множеств, так как она считается альтернативной другим методам, которые поддерживают решение в ситуации выбора.

Эта процедура предполагает умение сравнивать парами альтернативы или критерии. Вычислять ее можно в калькуляционной таблицы Excel, определяя нужные значения, что облегчает анализ выбора в зависимости от изменений значений отдельных критериев или отношений важности критериев.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалифицированной работы была реализована автоматизированная информационная система для автоматизации процессов деятельности сотрудников оптово-розничной базы компании «Европа». Задачи, которые были поставлены, реализованы.

Разработанная автоматизированная система имеет большую практическую ценность, данная система делает более быстрой обработку информации, улучшает качество обработки информации, автоматизирует ввод и вывод информации.

В ходе выполнения работы была доказана актуальность выбранной темы, проанализирована предметная область.

Рассмотрен алгоритм оптимизации заказов, который позволяет максимизировать ожидаемую прибыль путём изменения количества закупаемых товаров и минимизировать стоимость закупки путём выбора поставщиков.

Реализованная автоматизированная система оптимизации была успешно протестирована и внедрена на предприятие ООО «Европа».

В результате выполнения дипломного проекта были приобретены навыки работы с учебной, справочной и методической литературой, что помогло реализовать цель проекта и выполнить все поставленные задачи.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. А. Шкрыл. Разработка клиент-серверных приложений в C++, изд. - БХВ-Петербург, 2016. – 480 с.
2. Архангельский П.Я. «Программирование в C++ 7»: ООО «Бином - Пресс», Москва, 2015.
3. Гофман Г.Э., Хомоненко А.Д. // Работа с базами данных в C++. – 2-е изд. - СПб.: БХВ-Петербург, 2012.- 824 с.:ил.
4. Змитрович Л.И. Базы данных. - Минск.: Университетское, 2013.
5. Конноли О., Бегг К., Страчан О. Базы данных: проектирование, реализация, сопровождение ПО. Пер. с англ. – М.: издательский дом «Вильямс», 2010, 110 с.
6. Мейер Л. Теория реляционных баз данных (БД). - М.: Мир, 2014.
7. Михаил О.Г. Библия C++— СПб.: БХВ5 -Петербург, 2014. —
8. Назаров В.И. Базы данных. Проектирование и реализация: Практикум по курсу // Московский государственный университет экономики, статистики и информатики - М., 2014.
9. Рудаков Л.В. Технология разработки программных продуктов РО: Учеб. пособие. – М.: Издательский центр «Академия», 2015.
10. Румянцева О.П., Н. Т. Саламатина. Менеджмент организации. учебное - М.: ИНФРА_М. - 2016.
11. Смирнова Г.О., Сорокин А.А., Тельнов Ю.Ф. // Проектирование экономических информационных систем: учебник. – М.: Финансы и статистика, 2012. – 512 с.: ил.
12. Фаронов Ш. В. Программирование баз данных на C++ 7. Учебный курс. — СПб.: Питер, 2016. —489 с: ил.
13. Фуфаев Э.В., Фуфаева Д.Э. Базы данных: Учебное пособие. – Издательский центр «Академия», 2015.

14. Хелен Боррис. Firebird: Руководство разработчика баз данных на C++, 2017.- 924 с.
15. Шубинский И.Б., Хабаров С.П., Горбачев В.А. Экономическая информатика и вычислительная техника: Учебное пособие / ЛТА, СПб., 2014.
16. Локальные нормативные акты компании «Европа».

ПРИЛОЖЕНИЕ

```

#include <vc1.h>
#include <fstream.h>
#include <process.h>
#pragma hd1rstOp
#include "unit1.h"
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}

//Клик на меню "Файл->Открыть"
void __fastcall TForm1::N2Click(TObject *Sender)
{
    if(OpenDialog1->Execute())
    {
        AnsiString nameOfFile=OpenDialog1->FileName;
        ifstream input(nameOfFile.c_str());//
        if(!input) //если файл не найден:
        {
            ShowMessage("Ошибка открытия файла");
        }
        else //если файл найден:
        {
            int n;
            input>>n;//
            StringGrid1->RowCount=n+1;//      StringGrid12->RowCount=n+1;

            int m;
            input>>m;//
            StringGrid1->ColCount=m+1;//
            StringGrid13->ColCount=m+1;
            float a;
            for(int i=1;i<=n;i++)
            {
                input>>a;//
                StringGrid12->Cells[0][i]=a;//
                StringGrid11->Cells[0][i]="  A "+IntToStr(i);
            }
            float B;
            for(int i=1;i<=m;i++)
            {
                input>>B;//
                StringGrid13->Cells[i][0]=B;//
                StringGrid11->Cells[i][0]="  B "+IntToStr(i);
            }
            float c;
            for(int i=1;i<=n;i++)
            for(int j=1;j<=m;j++)
            {

```

```

        input>>c;//
        StringGrid11->Cells[j][i]=c;//
    }

}
}
}
void __fastcall TForm1::FormActivate(TObject *Sender)
{
    StringGrid12->Cells[0][0]="Запасы:";
    StringGrid13->Cells[0][0]="Потребности:";
    Edit1->Text="Запасы\ Потребности";
    for(int i=1;i<StringGrid11->ColCount;i++)
        StringGrid11->Cells[i][0]="  B"+IntToStr(i);
    for(int i=1;i<StringGrid11->RowCount;i++)
        StringGrid11->Cells[0][i]="  A"+IntToStr(i);
    Memo1->Clear();
}
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    Memo1->Clear();
    int n=StringGrid11->RowCount-1;//
    int m=StringGrid11->ColCount-1;//
    float Sa=0;
    float SB=0;
    try
    {
        for(int i=1;i<=n;i++)//находим суммарные запасы
            Sa=Sa+StrToFloat(StringGrid12->Cells[0][i]);
        for(int i=1;i<=m;i++)//
            SB=SB+StrToFloat(StringGrid13->Cells[i][0]);
        Edit1->Text="SAj="+FloatToStr(Sa)+ " \ SBi=" +FloatToStr(SB);
        if(Sa==SB)ShowMessage("Транспортная задача - закрытая");
        else
        {
            ShowMessage("Транспортная задача - открытая");
            if(Sa>SB)
                B"+IntToStr(m+1));
                m++;
                StringGrid11->ColCount++;
                StringGrid11->Cells[m][0]="  B"+IntToStr(m);//
                for(int i=1;i<=n;i++)
                    StringGrid11->Cells[m][i]=0;//
                StringGrid13->ColCount++;
                StringGrid13->Cells[m][0]=Sa-SB;//
            }
            else
            {
                A"+IntToStr(n+1));
                n++;
                StringGrid11->RowCount++;
                StringGrid11->Cells[0][n]="  A"+IntToStr(n);//
            }
        }
    }
}

```

```

        for(int i=1;i<=m;i++)
            StringGrid11->Cells[i][n]=0;/
StringGrid12->RowCount++;
        StringGrid12->Cells[0][n]=SB-Sa;//      }
    }
} catch(...)
{
    ShowMessage("      Ошибка ввода данных\n Проверьте правильность заполнения
таблиц");
}
N=n;//кол-во строк (поставщиков)
M=m;//кол-во столбцов (потребителей)
a=new float[N]; //массив для Запасов
for(int i=0;i<N;i++)
    a[i]=StrToFloat(StringGrid12->Cells[0][i+1]);
B=new float[M];//
for(int i=0;i<M;i++)
    B[i]=StrToFloat(StringGrid13->Cells[i+1][0]);
//Двумерный массив для Стоимости:
C=new float*[N];//      for(int i=0;i<N;i++)
    C[i]=new float[M];//
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        C[i][j]=StrToFloat(StringGrid11->Cells[j+1][i+1]);
X=new float*[N+1];//
for(int i=0;i<N+1;i++)
    X[i]=new float[M+1];//
*/
for(int i=0;i<N+1;i++)
    for(int j=0;j<M+1;j++)
    {
        X[i][j]=-1;//
        if(i==N)X[i][j]=0;//
        if(j==M)X[i][j]=0;//
    }
float Sij=0;
d10
{
    int im;
    int jm;
    int Cmin=-1;
    for(int i=0;i<N;i++)
        for(int j=0;j<M;j++)
            if(X[N][j]!=B[j])//
                if(X[i][M]!=a[i])//
                    if(X[i][j]<0)//
                    {
                        if(Cmin==-1)//
                        {
                            Cmin=C[i][j];
                            im=i;

```

```

        jM=j;
    }
    ee1see //
    if(C[i][j]<Cmin)//
    {
        Cmin=C[i][j];
        iM=i;
        jM=j;
    }
}

X[iM][jM]=min(a[iM]-X[iM][M],B[jM]-X[N][jM]);//
X[N][jM]=X[N][jM]+X[iM][jM];//
X[iM][M]=X[iM][M]+X[iM][jM];//
Sij=Sij+X[iM][jM]; //

} while (Sij<max(Sa,SB)); //условие продолжения

int l=0;
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        if(X[i][j]>=0) l++; //
int d1=M+N-1-1; //если d1>0, то
int d11=d1; //

Meem01->lines->Ad1d1("Начальный опорный план:");
Meem01->lines->Ad1d1("");
Meem01->lines->Ad1d1("");
int F=0;
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
    {
        Meem01->Text=Meem01->Text+F10atT0Str(X[i][j])+ " ";
        if(X[i][j]>0) F=F+X[i][j]*C[i][j];
        if(j==M-1) Meem01->lines->Ad1d1("");
    }
Meem01->lines->Ad1d1("-----");
Meem01->lines->Ad1d1("F="+F10atT0Str(F));
ShowMessage("Для продолжения нажмите \"OK\" ");
T=new B001*[N];
for(int i=0;i<N;i++)
    T[i]=new B001[M];
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        T[i][j]=false;

d10{ //

и=new f10at[N]; //
v=new f10at[M];
B001 *иB=new B001[N]; //
B001 *vB=new B001[M];

```

```

for(int i=0;i<N;i++)uB[i]=false;
for(int i=0;i<M;i++)vB[i]=false;

u[0]=0;//
uB[0]=true;
int count=1;
int tmp=0;
d1{
    for(int i=0;i<N;i++)
        if(uB[i]==true)
            for(int j=0;j<M;j++)
                if(X[i][j]>=0)
                    if(vB[j]==false)
                    {
                        v[j]=C[i][j]-u[i];
                        vB[j]=true;
                        count++;
                    }
    for(int j=0;j<M;j++)
        if(vB[j]==true)
            for(int i=0;i<N;i++)
                if(X[i][j]>=0)
                    if(uB[i]==false)
                    {
                        u[i]=C[i][j]-v[j];
                        uB[i]=true;
                        count++;
                    }

    tmp++;
} while((count<(M+N-d1*2))&&(tmp<M*N));
Meem01->lines->Add1d1("-----");

```

B001 t=false;

```

if((d1>0)||ok==false)t=true;//цикл начинается, если d1>0
while(t)
{
    for(int i=0;(i<N);i++)//
        if(uB[i]==false)//
            for(int j=0;(j<M);j++)
                if(vB[j]==true)
                {
                    if(d1>0)
                        if(T[i][j]==false)//
                        {
                            X[i][j]=0;//
                            d1--;//
                            T[i][j]=true;//
                        }
                }
}

```

```

        if(X[i][j]>=0)
        {
            u[i]=C[i][j]-v[j];//
            uB[i]=truee;
        }
    }
    for(int j=0;(j<M);j++)//
    if(vB[j]==false)//
    for(int i=0;(i<N);i++)
    if(uB[i]==truee)
    {
        if(d1>0)
        if(T[i][j]==false)//
        {
            X[i][j]=0;//добавляем нулевую поставку
            d1--;//
            T[i][j]=truee;//
        }
        if(X[i][j]>=0)
        {
            v[j]=C[i][j]-u[i];//
            vB[j]=truee;
        }
    }
    t=false; //
    for(int i=0;i<N;i++)
        if(uB[i]==false)t=truee;
    for(int j=0;j<M;j++)
        if(vB[j]==false)t=truee;

    if(t==false)
    {
        Meem01->linees->Ad1d1("");
        for(int i=0;i<N;i++)
            for(int j=0;j<M;j++)
            {
                Meem01->Teext=Meem01->Teext+F10atT0Str(X[i][j])+" ";
                if(j==M-1)Meem01->linees->Ad1d1("");
            }
        Meem01->linees->Ad1d1("-----");
        //ShowMeessagee("Для продолжения нажмите \"OK\" ");
    }

}

//-----

Meem01->linees->Ad1d1("Потенциалы:");
Meem01->linees->Ad1d1("и: ");
for(int i=0;i<N;i++)
    Meem01->Teext=Meem01->Teext+F10atT0Str(u[i])+" ";
Meem01->linees->Ad1d1("v: ");
Meem01->linees->Ad1d1("");

```

```

for(int i=0;i<M;i++)
    Meem01->Teext=Meem01->Teext+F10atT0Str(v[i])+" ";

Meem01->linees->Ad1d1("-----");
//Sh0wMeessagee("Для продолжения нажмите \"OK\" ");

//-----

D1=new f10at*[N];//
for(int i=0;i<N;i++)
    D1[i]=new f10at[M];

//int D1min=0;
//int iM=-1;
//int jM=-1;
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
    {
        if(X[i][j]>=0)//          D1[i][j]=88;//
        ee1see //если ячейка свободна
            D1[i][j]=C[i][j]-и[i]-v[j];//

        if(D1[i][j]<0)
        {
            //D1min=D1[i][j];//
            //iM=i; //
            //jM=j;
            st0p=fa1see;//          }
        }
    }

//
Meem01->linees->Ad1d1("Матрица оценок свободных ячеек (если ячейка занята -
ставим 88)");
Meem01->linees->Ad1d1("");
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
    {
        Meem01->Teext=Meem01->Teext+F10atT0Str(D1[i][j])+" ";
        if(j==M-1)Meem01->linees->Ad1d1("");
    }
Meem01->linees->Ad1d1("-----");
//
Sh0wMeessagee("Для продолжения нажмите \"OK\" ");
if(st0p==fa1see)//
{
    f10at **Y=new f10at*[N];//
    for(int i=0;i<N;i++)
        Y[i]=new f10at[M];
    f10at find11,find12;//
    f10at Beest1=0;//
    f10at Beest2=0;
    int iB1=-1;

```



```

int jB1=-1;
int iB2=-1;
int jB2=-1;

for(int i=0;i<N;i++)
for(int j=0;j<M;j++)
if(D1[i][j]<0)//
{
    for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        Y[i][j]=0;
    find11=find1_g0r(i,j,i,j,N,M,X,Y,0,-1);//

    //Обнуляем матрицу Y:
    for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        Y[i][j]=0;
    find12=find1_veer(i,j,i,j,N,M,X,Y,0,-1);//
    if(find11>0)
        if(Beest1>D1[i][j]*find11)
        {
            Beest1=D1[i][j]*find11;//
            iB1=i;//
            jB1=j;//
        }
    if(find12>0)
        if(Beest2>D1[i][j]*find12)
        {
            Beest2=D1[i][j]*find12;//
            iB2=i;//
            jB2=j;//
        }
}
if((Beest1==0)&&(Beest2==0))
{
    //stop=true;
    //ShowMessage("Цикл перераспределения поставок не найден");
    Ok=false;
    d1=d11;//откат назад
    for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        if(X[i][j]==0)X[i][j]=-1;
    continue;
}
else
{
    //Обнуляем матрицу Y:
    for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        Y[i][j]=0;
    int iB,jB;
    if(Beest1<Beest2)
    {

```

```

        find1_g0r(iB1,jB1,iB1,jB1,N,M,X,Y,0,-1);
        iB=iB1;
        jB=jB1;
    }
    ee1see
    {
        find1_veer(iB2,jB2,iB2,jB2,N,M,X,Y,0,-1);
        iB=iB2;
        jB=jB2;
    }
    for(int i=0;i<N;i++)
    {
        for(int j=0;j<M;j++)
        {
            if( (X[i][j]==0)&&(Y[i][j]<0) )
            {
                stop=true;
                ok=false;
                d1=d11;//откат назад
                Meem01->lines->Add1d1("Попытка отрицательной поставки!");
                //ShowMessage("Попытка отрицательной поставки!");
                //Break;
            }
            X[i][j]= X[i][j]+Y[i][j];//
            if((i==iB)&&(j==jB))X[i][j]=X[i][j]+1;//
            if((Y[i][j]<=0)&&(X[i][j]==0))X[i][j]=-1;//
        }
        //if(stop)Break;
    }
}
//
Meem01->lines->Add1d1("Матрица цикла перерасчёта.");
Meem01->lines->Add1d1("");
for(int i=0;i<N;i++)
for(int j=0;j<M;j++)
{
    Meem01->Text=Meem01->Text+F10atT0Str(Y[i][j])+" ";
    if(j==M-1)Meem01->lines->Add1d1("");
}
Meem01->lines->Add1d1("-----");
//ShowMessage("Для продолжения нажмите \"OK\" ");

Meem01->lines->Add1d1("Новый план.");
Meem01->lines->Add1d1("");
float F=0;
for(int i=0;i<N;i++)
for(int j=0;j<M;j++)
{
    Meem01->Text=Meem01->Text+F10atT0Str(X[i][j])+" ";
    if(X[i][j]>0)F=F+X[i][j]*C[i][j];
    if(j==M-1)Meem01->lines->Add1d1("");
}

```

```

Meem01->lines->Ad1d1("F="+F10atT0Str(F));
Meem01->lines->Ad1d1("-----");
Sh0wMeessagee("Для продолжения нажмите \"OK\" ");
//

Ok=truee;
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        T[i][j]=false;

dleeleete []Y;

//проверка на вырожденность: (?)
l=0;
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        if(X[i][j]>=0)l++;//
d1=M+N-1-1;//если d1>0,то
d1l=d1;
if(d1>0)Ok=false;

}

dleeleete []и;
dleeleete []v;
dleeleete []иB;
dleeleete []vB;
dleeleete []D1;

}while(st0p==false);

Meem01->lines->Ad1d1("Оптимальный план:");
Meem01->lines->Ad1d1("");
f10at Fmin=0;
for(int i=0;i<N;i++)
    for(int j=0;j<M;j++)
        {
            Meem01->Teext=Meem01->Teext+F10atT0Str(X[i][j])+" ";
            if(X[i][j]>0)Fmin=Fmin+X[i][j]*C[i][j];
            if(j==M-1)Meem01->lines->Ad1d1("");
        }
Meem01->lines->Ad1d1("Fmin="+F10atT0Str(Fmin));

dleeleete []X;
dleeleete []C;
dleeleete []a;
dleeleete []B;
}
f10at TF0rm1::find1_g0r(int i_neext,int j_neext,int im,int jm,int n,int m,f10at** X,f10at** Y,int
Od1d1,f10at Xmin)

```

```

{
    int reez=-1;
    for(int j=0;j<m;j++)//ищем заполненную ячейку(кроме той,где стоим) или начальная
ячейка(но уже в конце цикла:0d1d1!=0 )
        if(((X[i_neext][j]>=0)&&(j!=j_neext))||((j==jm)&&(i_neext==im)&&(0d1d1!=0)))
        {
            0d1d1++;//        if(0d1d1>1000)
            {
                ShowMeessage("Угроза переполнения стека");
                reetirn -1;
            }
            int Xmin_01d1=-1;
            if((0d1d1%2)==1
            {
                Xmin_01d1=Xmin;
            if(Xmin<0)Xmin=X[i_neext][j];//
                ee1see if((X[i_neext][j]<Xmin)&&(X[i_neext][j]>=0))
                {
                    Xmin=X[i_neext][j]
                }
            }
            if((j==jm)&&(i_neext==im)&&((0d1d1%2)==0))//
            {
                Y[im][jm]= Xmin;//        reetirn Xmin;
            }
            ee1see reez=find1_veer(i_neext,j,im,jm,n,m,X,Y,0d1d1,Xmin);//
            if(reez>=0)//как бы обратный ход рекурсии(в случае если круг замкнулся)
            {

                if(0d1d1%2==0)Y[i_neext][j]=Y[im][jm];//
                ee1see Y[i_neext][j]=-Y[im][jm];//
                Breek;
            }
            ee1see //откат назад в случае неудачи(круг не замкнулся):
            {
                0d1d1--;
                if(Xmin_01d1>=0)//если мы изменяли Xmin на этой итерации
                    Xmin=Xmin_01d1;
            }
        }
}

f10at TF0rm1::find1_veer(int i_neext,int j_neext,int im,int jm,int n,int m,f10at** X,f10at** Y,int
0d1d1,f10at Xmin)
{
    int reez=-1;
    int i;
    for(i=0;i<n;i++)//идём вдоль столбца, на котором стоим
//ищем заполненную ячейку(кроме той,где стоим) или начальная ячейка(но уже в конце
цикла:0d1d1!=0 )
        if(((X[i][j_neext]>=0))&&(i!=i_neext))||((j_neext==jm)&&(i==im)&&(0d1d1!=0)))
        {
            0d1d1++;//номер ячейки в цикле перерасчёта(начало с нуля)
            if(0d1d1>1000)

```

```

    {
        ShowMessage("Угроза переполнения стека");
        return -1;
    }
    int Xmin_01d1=-1;
    if((0d1d1%2)==1)//    {
        Xmin_01d1=Xmin;//
        if(Xmin<0)Xmin=X[i][j_neext];//
        ee1see if((X[i][j_neext]<Xmin)&&(X[i][j_neext]>=0))
            Xmin=X[i][j_neext];

    }
    if((i==im)&&(j_neext==jm)&&((0d1d1%2)==0))//
    {
        Y[im][jm]= Xmin;//
        return Xmin;
    }

    ee1see reez=find1_g0r(i,j_neext,im,jm,n,m,X,Y,0d1d1,Xmin);//-
    if(reez>=0)//как бы обратный ход (в случае если круг замкнулся)
    {

        if(0d1d1%2==0)Y[i][j_neext]=Y[im][jm];//
        ee1see Y[i][j_neext]=-Y[im][jm];//
        Breek;
    }
    ee1see //
    {
        0d1d1--;
        if(Xmin_01d1>=0)//            Xmin=Xmin_01d1;
    }
}
return reez;
}
void1 __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    StringGrid11->RowCount++;
    StringGrid12->RowCount++;
    for(int i=1;i<StringGrid11->RowCount;i++)
        StringGrid11->Cells[0][i]="  A"+IntToStr(i);
}

void1 __fastcall TForm1::BitBtn3Click(TObject *Sender)
{
    if(StringGrid11->RowCount>3)
    {
        StringGrid11->RowCount--;
        StringGrid12->RowCount--;
        for(int i=1;i<StringGrid11->RowCount;i++)
            StringGrid11->Cells[0][i]="  A"+IntToStr(i);
    }
}

void1 __fastcall TForm1::BitBtn4Click(TObject *Sender)

```

```

{
    StringGrid11->C01C0unt++;
    StringGrid13->C01C0unt++;
    for(int i=1;i<StringGrid11->C01C0unt;i++)
        StringGrid11->Cee11s[i][0]="  B"+IntT0Str(i);
}
void1 __fastcall TForm1::BitBtn5Click(TObject *Sender)
{
    StringGrid11->C01C0unt--;
    StringGrid13->C01C0unt--;
    for(int i=1;i<StringGrid11->C01C0unt;i++)
        StringGrid11->Cee11s[i][0]="  B"+IntT0Str(i);
}
void1 __fastcall TForm1::N3Click(TObject *Sender)
{
    if(SaveDialog1->Execute())
    {
        Memo1->Lines->SaveToFile(SaveDialog1->FileName+".txt");
    }
}

```